

# Construcción, simulación y programación de un robot cuadrúpedo multipropósito de código abierto



Máster Universitario en Automática  
y Robótica

## Trabajo Fin de Máster

Autor:

Jonathan Mortes Esquer

Tutor:

Carlos Alberto Jara Bravo

Julio 2021



Universitat d'Alacant  
Universidad de Alicante





# Construcción, simulación y programación de un robot cuadrúpedo multipropósito de código abierto

---

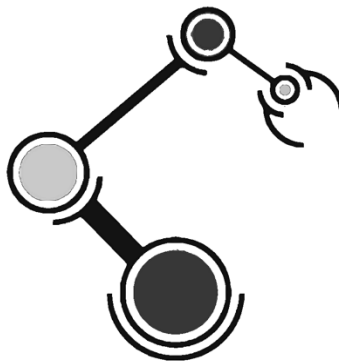
## **Autor:**

Jonathan Mortes Esquer

## **Tutor:**

Carlos Alberto Jara Bravo

*Departamento de física, ingeniería de sistemas y teoría de la señal*



Máster en Automática y Robótica



Escuela  
Politécnica  
Superior



Universitat d'Alacant  
Universidad de Alicante

Alicante, junio 2021

*Dedicado a mi abuelo, Boanerges,  
por haber creído siempre en mí.*



## Resumen

En este proyecto se pretende modelar, simular y construir un robot cuadrúpedo basado en la plataforma open source Spot Micro (robot inspirado en Spot mini de Boston Dynamics). El objetivo es mediante el estudio de los distintos sistemas actualmente disponibles, desarrollar una plataforma robótica de bajo coste multipropósito, por fases de desarrollo. Se explorará el sistema actual, se corregirán los defectos y finalmente se mejorará dotándolo con la posibilidad de alcanzar cierto nivel de autonomía. Se pretende también generar un programa de gestión que permita el control de sus articulaciones y de los distintos sensores incorporados así como establecer las bases para el entrenamiento de un agente con aprendizaje por refuerzo que sirva como futuro controlador del robot vía teleoperación o control manual.



## **Preámbulo: motivación y justificación**

La construcción y operación de un robot cuadrúpedo surgió en un primer momento a modo de reto personal, con el fin de poner en práctica todos los conocimientos obtenidos en el grado de ingeniería robótica. Más tarde, el tutor de este trabajo me hizo ver que la complejidad de esta tarea podría ser suficiente para realizarla en un Trabajo Fin de Máster. La idea inicial de desarrollar un robot en mis ratos libres se convirtió en un proyecto estructurado, con objetivos y tiempos definidos que encajase como un trabajo final de máster. Gracias al apoyo del tutor, he podido desarrollarlo y llevar el proyecto mucho más allá de lo que originalmente había concebido y, gracias al hecho de haberlo incorporado a la parte curricular del máster a modo de TFM, he podido desarrollarlo con mayor profundidad.



# Agradecimientos

Antes de comenzar, me gustaría agradecerle en estas breves líneas a aquellas personas que han aportado un poco de su valioso tiempo en pos de ayudarme a realizar este trabajo.

En primer lugar, me gustaría agradecerle a mi tutor, Carlos A. Jara su orientación, guía y apoyo constante, para que aquella idea que empezó como un reto y hobby personal y que terminara siendo, con un poco de enfoque y adaptación, este trabajo final de máster. Has estado desde el principio apoyándome, solucionando mis dudas, motivándome y orientándome, no solo a nivel académico o profesional, sino también a nivel personal, asegurándote de que, ante todo, siempre pudiese avanzar y crecer.

También me gustaría agradecerle especialmente a Margarita Conesa, su apoyo y motivación constante desde la concepción de la idea. Sé que he dado muchas vueltas, en parte por mi afán por el perfeccionismo y adición de más y más ideas conforme me iban surgiendo. Te agradezco haberme ayudado a organizar mis prioridades y a garantizar que todo mi trabajo quedase bien redactado y estructurado. Sin ti no hubiera sido posible.

Por último, a mi familia, por haberme apoyado a lo largo y ancho de mi trayectoria académica y sin los que no sería la persona que soy hoy. Han sido momentos duros e intensos, pero nunca dejasteis de animarme a seguir y a sacar lo mejor de mí. Por todo ello, os estoy inmensamente agradecido.

Y estoy seguro de que me dejo a muchas más personas que harían que la longitud de esta parte se equiparase al trabajo, pero quiero decir que me acuerdo de todos vosotros: familiares, profesores, mentores, compañeros, amigos y conocidos que habéis aportado vuestro granito en el viaje que me ha llevado hasta este momento y que seguro que seguiréis estando ahí más adelante.

A todos vosotros, de corazón, ¡muchísimas gracias por todo!

• Jonathan



*“Si algo es lo suficientemente importante,  
incluso si las probabilidades están en tu contra,  
deberías de seguir intentándolo.”*

*- Elon Musk*

*“En los buenos momentos, sé tu mejor versión.*

*En los malos, nunca la dejes atrás.”*

*- Jonathan Mortes Esquer.*



# Índice de contenidos

<b>ÍNDICE DE FIGURAS .....</b>	<b>16</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>21</b>
<b>CAPÍTULO 1: INTRODUCCIÓN .....</b>	<b>22</b>
1.1    OBJETIVOS .....	22
1.2    ESTRUCTURA .....	23
1.3    DOCUMENTACIÓN .....	24
<b>CAPÍTULO 2: PLANTEAMIENTO INICIAL DEL PROTOTIPO.....</b>	<b>25</b>
2.1    BÚSQUEDA DE PLATAFORMAS ROBÓTICAS DE CÓDIGO ABIERTO .....	25
2.2    ELECCIÓN Y PLANTEAMIENTO INICIAL DE LA ELABORACIÓN DEL PROTOTIPO .....	30
2.3    ANÁLISIS DE CAPACIDADES, CARACTERÍSTICAS Y PLANTEAMIENTO DE LOS OBJETIVOS A REALIZAR.....	31
2.4    ANÁLISIS DE LOS REQUISITOS PARA LLEVAR A CABO EL PROTOTIPO .....	32
2.4.1 <i>Información visual del estado del robot</i> .....	32
2.4.2 <i>Control preciso de las articulaciones</i> .....	32
2.4.3 <i>Control del estado y autonomía de la batería</i> .....	33
2.4.4 <i>Movimiento natural/inteligente</i> .....	34
2.4.5 <i>Simulación del robot</i> .....	34
2.4.6 <i>Mejora de la sensorización</i> .....	34
2.4.7 <i>Monitorización externa</i> .....	35
<b>CAPÍTULO 3: ESTADO DEL ARTE.....</b>	<b>36</b>
3.1.    BREVE HISTORIA E INTRODUCCIÓN A LA ROBÓTICA MÓVIL CUADRÚPEDA .....	36
3.2.    PLATAFORMAS ROBÓTICAS CUADRÚPEDAS ACTUALES.....	38
3.3.    SISTEMAS DE CONTROL DE ROBOTS .....	45
3.3.1 <i>Control directo sin controlador</i> .....	46
3.3.2 <i>Control con trayectorias generadas automáticamente sin controlador</i> .....	46
3.3.3 <i>Control directo con cinemática inversa</i> .....	47
3.3.4 <i>Control con curvas de Bézier y aprendizaje automático por refuerzo (RL)</i> .....	48
3.3.5 <i>Control con navegación autónoma (SLAM)</i> .....	49
3.4.    SENSORES Y ACTUADORES EN ROBOTS CUADRÚPEDOS.....	51
3.4.1 <i>Sensores de tiempo de vuelo</i> .....	51
3.4.2 <i>Sensores ópticos (sin ToF)</i> .....	52
3.4.3 <i>Sensores de presión/fuerza y posición/velocidad</i> .....	52
3.4.4 <i>Sensores de orientación</i> .....	53
3.4.5 <i>Sensores de sistemas</i> .....	53
3.4.6 <i>Motores y actuadores</i> .....	54
3.5.    SISTEMAS DE SIMULACIÓN Y DISEÑO.....	56
3.5.1 <i>Gazebo [41]</i> .....	56
3.5.2 <i>RViz [42]</i> .....	56
3.5.3 <i>Coppelia Sim/V-Rep [43]</i> .....	56
3.5.4 <i>CAD/CAE/CAM [44]</i> .....	57
3.5.5 <i>ADAMS [45]</i> .....	57
3.5.6 <i>Pybullet [46]</i> .....	57

3.5.7	<i>Matlab + Simulink [47]</i> .....	57
3.6	SISTEMAS Y MÉTODOS PARA LA TELEOPERACIÓN Y MONITORIZACIÓN .....	58
3.6.1	<i>Teleoperación clásica con mandos de control</i> .....	58
3.6.2	<i>Teleoperación avanzada con mandos de control</i> .....	58
3.6.3	<i>Teleoperación o control autónomo</i> .....	58
3.7	SISTEMAS Y TÉCNICAS DE IMPRESIÓN 3D .....	59
3.7.1	<i>Técnicas de impresión 3D</i> .....	59
3.7.2	<i>Selección del método y material de impresión</i> .....	63
3.7.3	<i>Configuraciones básicas recomendadas</i> .....	65
3.7.4	<i>Patrón interno de relleno</i> .....	67
<b>CAPÍTULO 4: METODOLOGÍA PARA EL DESARROLLO</b> .....		<b>71</b>
4.1	<i>Diseño e impresión del prototipo</i> .....	71
4.2	<i>Construcción y pruebas básicas del prototipo</i> .....	72
4.3	<i>Control básico del prototipo</i> .....	72
4.4	<i>Control avanzado, programación y simulación del prototipo</i> .....	73
<b>CAPÍTULO 5: IMPLEMENTACIÓN Y DESARROLLO DEL PROTOTIPO</b> .....		<b>74</b>
5.1	DISEÑO E IMPRESIÓN DEL PROTOTIPO: FASE 1 .....	76
5.1.1	<i>Obtención de modelos editables para Autodesk</i> .....	76
5.1.2	<i>Ensamblado y comprobación del modelo virtual</i> .....	77
5.1.3	<i>Estudio inicial de la cinemática del robot</i> .....	80
5.1.4	<i>Elección de los componentes electrónicos y sensores</i> .....	83
5.1.5	<i>Elaboración del esquema electrónico</i> .....	87
5.1.6	<i>Elaboración de los archivos de impresión</i> .....	88
5.4.4	<i>Elaboración de las tablas de costes</i> .....	90
5.1.8	<i>Impresión de las piezas estructurales, problemas y soluciones</i> .....	93
5.2	CONSTRUCCIÓN Y PRUEBAS BÁSICAS: FASE 1 .....	94
5.2.1	<i>Preparación del cableado eléctrico para el circuito principal y montaje</i> .....	94
5.2.2	<i>Elaboración de códigos de testeo y evaluación del movimiento de las articulaciones</i> .....	96
5.3	DISEÑO E IMPRESIÓN DEL PROTOTIPO: FASE 2 .....	97
5.3.1	<i>Planteamiento de mejoras al sistema</i> .....	97
5.3.2	<i>Estudio dinámico del sistema con CAD/CAE</i> .....	98
5.3.3	<i>Estudio de tensiones del sistema con CAD/CAE</i> .....	106
5.3.4	<i>Elección de los nuevos componentes electrónicos y sensores</i> .....	111
5.3.5	<i>Diseño de las nuevas piezas estructurales</i> .....	112
5.3.6	<i>Diseño del sistema de detección y medida de distancias mediante ultrasonidos (USDAR)</i> 114	
5.3.7	<i>Ensamblado y comprobación del modelo virtual</i> .....	121
5.3.8	<i>Revisión del esquema electrónico</i> .....	122
5.3.9	<i>Revisión de las tablas de costes e impresión</i> .....	124
5.4	CONSTRUCCIÓN Y VERIFICACIÓN: FASE 2 .....	126
5.4.1	<i>Cambios en el cableado, componentes y distribución interna</i> .....	126
5.4.2	<i>Elaboración de códigos de testeo</i> .....	127
5.4.4	<i>Pruebas de funcionamiento de sensores y electrónica</i> .....	128
5.4.3	<i>Montaje del robot</i> .....	130
5.4.5	<i>Evaluación del movimiento de las articulaciones, USDAR y componentes</i> .....	133
5.5	DISEÑO E IMPRESIÓN DEL PROTOTIPO: FASE 3 .....	134

5.5.1	<i>Planteamiento de mejoras al sistema .....</i>	134
5.5.2	<i>Elección de los nuevos componentes electrónicos y sensores .....</i>	136
5.5.3	<i>Diseño de las nuevas piezas del robot .....</i>	137
5.5.4	<i>Ensamblado y comprobación del modelo virtual.....</i>	143
5.5.5	<i>Revisión del esquema electrónico.....</i>	146
5.5.6	<i>Elaboración de los archivos de impresión y revisión de las tablas de costes .....</i>	147
5.5.7	<i>Impresión de las piezas estructurales.....</i>	149
5.6	<b>CONSTRUCCIÓN Y VERIFICACIÓN: FASE 3.....</b>	150
5.6.1	<i>Preparación de las piezas preexistentes.....</i>	150
5.6.2	<i>Preparación del nuevo cableado eléctrico interno .....</i>	152
5.6.3	<i>Elaboración de códigos de testeo y pruebas de funcionamiento.....</i>	155
5.6.4	<i>Calibrado de los motores.....</i>	157
5.6.5	<i>Montaje del robot.....</i>	158
5.6.6	<i>Construcción de una fuente de alimentación externa y cable de alimentación (adicional)</i> 165	
5.7	<b>CONTROL, PROGRAMACIÓN Y SIMULACIÓN: FASE 3.....</b>	166
5.7.1	<i>Preparación del entorno de programación.....</i>	167
5.7.2	<i>Comprobación del funcionamiento de los paquetes instalados .....</i>	167
5.7.3	<i>Elaboración del código de control básico, pruebas y correcciones del robot .....</i>	168
5.7.4	<i>Evaluación del control básico .....</i>	170
5.7.5	<i>Adaptación de los paquetes y códigos base a ROS Noetic y al prototipo .....</i>	175
5.7.6	<i>Verificación de la comunicación con ROS y mando remoto.....</i>	175
5.7.7	<i>Estudio del algoritmo GMBC y su modelo de aprendizaje por refuerzo.....</i>	178
5.7.8	<i>Entrenamiento del modelo de marcha sobre terreno irregular con aprendizaje por refuerzo y el algoritmo GMBC.....</i>	182
	<b>CAPÍTULO 6: CONCLUSIONES .....</b>	187
	<b>CAPÍTULO 7: TRABAJOS FUTUROS .....</b>	191
	<b>CAPÍTULO 8: BIBLIOGRAFÍA.....</b>	192
	<b>ANEXO 1: ENLACES A LOS CÓDIGOS, PROGRAMAS, MULTIMEDIA Y EXPLICACIONES .....</b>	200
	<b>ANEXO 2: AMPLIACIÓN DEL ESTADO DEL ARTE .....</b>	201
	<i>Sistemas de control de robots tradicionales.....</i>	201
	<i>Sistemas de control con algoritmos y redes neuronales aplicadas.....</i>	202
	<i>Sensores y actuadores en robots cuadrúpedos.....</i>	203
	<i>Sistemas de simulación y tele operación .....</i>	203
	<i>Bibliografía de los artículos de investigación arriba mencionados:.....</i>	203
	<b>ANEXO 3: IMPRESIÓN 3D .....</b>	212
	<b>ANEXO 4: ESTUDIO DINÁMICO DE UNA EXTREMIDAD .....</b>	213
	<b>ANEXO 5: ESTUDIO DE TENSIONES DE UNA EXTREMIDAD .....</b>	214

# Índice de figuras

FIGURA 1. ROBOT DUM-E DE LA SAGA IRON MAN DE MARVEL (IZQUIERDA)[1] Y ROBOT SPOT DE BOSTON DYNAMICS (DERECHA)[2].	25
FIGURA 2. ROBOT MOVEO [3].	26
FIGURA 3. ROBOT NIRYO ONE [4].	27
FIGURA 4. ROBOT HEXAPOD [5].	28
FIGURA 5. ROBOT KAME MINI [6].	29
FIGURA 6. ROBOT SPOT MICRO ORIGINAL [7].	29
FIGURA 7. ROBOTS BIGDOG (A) Y LS3 (B) DE BOSTON DYNAMICS [15].	39
FIGURA 8. ROBOTS SPOT CLASSIC (A) Y SPOT (B) DE BOSTON DYNAMICS [2][15].	40
FIGURA 9. ROBOTS CHETAH 3 (A) Y MINICHETAH (B) DEL MIT [16][18].	41
FIGURA 10. ROBOTS DE UNITREE ROBOTICS [19].	41
FIGURA 11. SOLUCIÓN ZERO POINT DE GHOST ROBOTICS [20].	42
FIGURA 12. ROBOT BIOINSPIRADO DE LA TMU[21].	43
FIGURA 13. ROBOT HYQ DEL INSTITUTO ITALIANO DE TECNOLOGÍA [24].	43
FIGURA 14. ROBOT RUSH [25].	44
FIGURA 15. ROBOT ANYMAL DE ANYBOTICS AG [26].	44
FIGURA 16. ROBOT PNEUPARD DE LA UNIVERSIDAD DE OSAKA [27].	45
FIGURA 17. ESQUEMA DE CONTROL CON CINEMÁTICA INVERSA (FUENTE PROPIA).	47
FIGURA 18. ENTRENAMIENTO POR REFUERZO DEL ROBOT OPEN QUADRUPEL [36].	49
FIGURA 19. SIMULACIÓN DE SPOT MICRO CON UN LIDAR EN RVIZ (COMPUESTO) [38].	50
FIGURA 20. SIMULACIÓN EN PYBULLET DEL PROYECTO Y ROBOT REAL [37].	50
FIGURA 21. COMPARATIVA DE IMPRESIÓN CON RESINA [49].	62
FIGURA 22. COMPARATIVA DE LOS MATERIALES PARA FDM [50].	64
FIGURA 23. COMPARATIVA DE DIVERSOS PATRONES DE RELLENO [51].	67
FIGURA 24. COMPARATIVA DE DENSIDAD DE RELLENO INTERNO [51].	68
FIGURA 25. COMPARATIVA DE FALLO A LA CARGA SEGÚN PATRÓN INTERNO Y ORIENTACIÓN [52].	68
FIGURA 26. COMPARATIVA DE RESISTENCIA A PRESIÓN SEGÚN PATRÓN INTERNO [53].	69
FIGURA 27. COMPARATIVA DEL TIEMPO DE IMPRESIÓN SEGÚN PATRÓN INTERNO [52].	69
FIGURA 28. COMPARATIVA GENERAL DE LA DENSIDAD DEL RELLENO FRENTE AL COSTE, CALIDAD, VELOCIDAD Y RESISTENCIA [54].	70
FIGURA 29. COMPARATIVA DE LA DENSIDAD DEL RELLENO FRENTE AL PESO Y EL TIEMPO DE IMPRESIÓN [54].	70
FIGURA 30. ESQUEMA DE CONTROL DEL SISTEMA PROPUESTO (FUENTE PROPIA).	74
FIGURA 31. ALGORITMO GMBC [34].	74
FIGURA 32. ESQUEMA DEL SISTEMA PROPUESTO (FUENTE PROPIA).	75
FIGURA 33. IMPORTACIÓN DEL ARCHIVO “WRIST.STL” ORIGINAL EN AUTODESK (FUENTE PROPIA).	77
FIGURA 34. EXPORTACIÓN DEL ARCHIVO “WRIST.FCSTD” ORIGINAL AL FORMATO “STEP” (FUENTE PROPIA).	77
FIGURAS 35 Y 36. PIE ENSAMBLADO (IZQUIERDA) Y FRONTAL ENSAMBLADO (DERECHA) (FUENTE PROPIA).	78
FIGURAS 37 Y 38. PARTE TRASERA ENSAMBLADA (V2) (IZQUIERDA) Y CUERPO ENSAMBLADO (DERECHA) (FUENTE PROPIA).	78
FIGURA 39. ROBOT ENSAMBLADO ORIGINAL [8].	79
FIGURA 40. ENSAMBLAJE EN INVENTOR DE LA FASE 0 (FUENTE PROPIA).	79
FIGURA 41. PRUEBAS DE MOVIMIENTO DEL MODELO DE LA FASE 1 (FUENTE PROPIA).	80
FIGURA 42. MEDIDAS DEL CUERPO DEL ROBOT (FUENTE PROPIA).	80
FIGURA 43. MEDIDAS DEL HOMBRO DEL ROBOT (FUENTE PROPIA).	81
FIGURA 44. MEDIDAS DEL MIEMBRO SUPERIOR DEL ROBOT (FUENTE PROPIA).	81
FIGURA 45. MEDIDAS DEL MIEMBRO INFERIOR DEL ROBOT (FUENTE PROPIA).	81

FIGURA 46. CADENAS CINEMÁTICAS DE LAS PATAS DE SPOT MICRO (FUENTE PROPIA).	82
FIGURAS 47 Y 48. CADENAS CINEMÁTICAS DE LAS PATAS DE SPOT MICRO (FUENTE PROPIA).	83
FIGURA 49. DIAGRAMA ELECTRÓNICO DE CONEXIONES PARA LA FASE 1 (FUENTE PROPIA).	88
FIGURA 50. VISTA TRAS LA SEGMENTACIÓN DE LA PLACA DEL CIRCUITO DE LA FASE 3 (FUENTE PROPIA).	89
FIGURA 51. IMPRESIÓN DE UN MIEMBRO SUPERIOR CON VISTA DEL PATRÓN GIROIDE (FUENTE PROPIA).	93
FIGURA 52. IMPRESIÓN DE UNA PATA INFERIOR DE LA FASE 3 CON EL SISTEMA DE CALIBRADO Y NIVELACIÓN AUTOMÁTICA (FUENTE PROPIA).	94
FIGURA 53. COMPONENTES SOBRE EL CIRCUITO DE LA FASE 1 (FUENTE PROPIA).	95
FIGURA 54. PROTOTIPO DE LA FASE 1 (FUENTE PROPIA).	95
FIGURA 55. ROTURA A TORSIÓN DEL SERVOHORN DELANTERO IZQUIERDO (FUENTE PROPIA).	97
FIGURA 56. ILUSTRACIÓN DE LAS MEDIDAS DESEADAS (FUENTE PROPIA).	99
FIGURA 57. MOVIMIENTO IMPUESTO AL GRADO DE LIBERTAD CORRESPONDIENTE AL HOMBRO FLEXOR/EXTENSOR (FUENTE PROPIA).	99
FIGURA 58. MOVIMIENTO IMPUESTO AL GRADO DE LIBERTAD CORRESPONDIENTE AL HOMBRO ABDUCTOR Y AL CODO (FUENTE PROPIA).	100
FIGURA 59. CONFIGURACIÓN DEL ANÁLISIS DINÁMICO INSTANTES ANTES DE SU EJECUCIÓN (FUENTE PROPIA).	100
FIGURA 60. RESULTADOS DEL ANÁLISIS DINÁMICO PARA EL HOMBRO FLEXOR/EXTENSOR (FUENTE PROPIA).	102
FIGURA 61. RESULTADOS DEL ANÁLISIS DINÁMICO PARA EL HOMBRO ABDUCTOR (FUENTE PROPIA).	103
FIGURA 62. RESULTADOS DEL ANÁLISIS DINÁMICO PARA EL HOMBRO ABDUCTOR (FUENTE PROPIA).	104
FIGURA 63. RESULTADOS DEL ANÁLISIS DINÁMICO PARA EL HOMBRO ABDUCTOR CON UNA CARGA DE 30N (FUENTE PROPIA).	105
FIGURA 64. EJEMPLO 1 DE LOS RESULTADOS DEL ANÁLISIS ESTÁTICO DE TENSIONES DISTRIBUIDAS (FUENTE PROPIA).	107
FIGURA 65. EJEMPLO 1 DE LOS RESULTADOS DEL ANÁLISIS ESTÁTICO DE TENSIONES DISTRIBUIDAS (FUENTE PROPIA).	110
FIGURA 66. CIRCUITO V2 (FUENTE PROPIA).	112
FIGURA 67. COBERTURA TRASERA (IZQUIERDA) Y MONTURA PARA SENSORES DE CORRIENTE, TENSIÓN Y BOTÓN (DERECHA) (FUENTE PROPIA).	113
FIGURA 68. TAPA SUPERIOR TRASERA V2 (FUENTE PROPIA).	114
FIGURA 69. PRIMER BOCETO DEL USDAR Y LOS RANGOS DE VISIÓN DEL ROBOT (FUENTE PROPIA).	115
FIGURA 70A. BOCETO DEL HW SUPERIOR DEL USDAR. EN LA PARTE SUPERIOR DISPONEMOS DE LA VISTA FRONTAL. EN LA IZQUIERDA TENEMOS EL MOTOR EN EL CENTRO CON EL EJE DE ACOPLE (NEGRO) Y EL SERVOHORN (AMARILLO) JUNTO CON LOS SENSORES DE ULTRASONIDOS. A LA DERECHA SE ENCUENTRA LA VISTA SUPERIOR CON EL ANILLO (AMARILLO) DE LEDS ANCLADO (FUENTE PROPIA).	116
FIGURA 70B. REPRESENTACIÓN TRIDIMENSIONAL DE LA FIGURA 70A (FUENTE PROPIA).	116
FIGURA 71. MODELO DE LA BASE DE MONTAJE (SERVORADARBASE_v14.STEP) (FUENTE PROPIA).	117
FIGURA 72. MODELO DE LA TAPA SUPERIOR FRONTAL (TOP_FRONT_COVER_v2.STEP) (FUENTE PROPIA).	118
FIGURA 73. MODELO DEL NÚCLEO DE SOPORTE (NUCLEO_v9.STEP). A LA IZQUIERDA LA VISTA SUPERIOR, EN EL CENTRO LA CUBIERTA PARA LA TUERCA M3 Y A LA DERECHA EL ENGANCHE DEL SERVOHORN (FUENTE PROPIA).	119
FIGURA 74. MODELO DE LA PIEZA DE SOPORTE DE LOS SENSORES DE ULTRASONIDOS (US_SUPPORT_v6.STEP) (FUENTE PROPIA).	119
FIGURA 75. A LA IZQUIERDA EL MODELO DEL PERNO Y REMACHE (PERNOS_v6.STEP) Y A LA DERECHA EL MODELO DE LA TAPA FRONTAL DEL SENSOR DE (US_FRONT_v4.STEP) (FUENTE PROPIA).	120
FIGURA 76. MODELO DE LA TAPA SUPERIOR (TOPRADAR_v19.STEP) (FUENTE PROPIA).	121
FIGURA 77. MODELO DEL USDAR EN SU VERSIÓN 2ª ENSAMBLADO EN VISTA SUPERIOR E INFERIOR (FUENTE PROPIA).	122
FIGURA 78. MODELO DEL USDAR EN SU VERSIÓN 2ª ENSAMBLADO EN VISTA LATERAL (FUENTE PROPIA).	122
FIGURA 79A. DIAGRAMA DE CIRCUITO V2 (FUENTE PROPIA).	123
FIGURA 79B. DIAGRAMA DE CIRCUITO V3 (FUENTE PROPIA).	124
FIGURA 80. CIRCUITO INTERRUPTOR CON SENSORES V2 (FUENTE PROPIA).	126

FIGURA 81. CIRCUITO PRINCIPAL ENSAMBLADO V2 (FUENTE PROPIA).....	127
FIGURA 82. VERIFICACIÓN DE TENSIONES EN EL CIRCUITO PRINCIPAL V2 (FUENTE PROPIA).....	129
FIGURA 83. A LA IZQUIERDA LA PLACA ARDUINO EMPLEADA PARA LA VERIFICACIÓN DE LOS SERVOMOTORES Y AGRUPACIÓN EN GRUPOS (CENTRO) CON LAS PATAS DE LA FASE 1 A LA DERECHA. (FUENTE PROPIA).....	129
FIGURA 84. COMPUESTO DE DISTINTAS PRUEBAS DE SOPORTE DE LA TAPA SUPERIOR CON EL NERVIOS CENTRAL (FUENTE PROPIA).....	130
FIGURA 85A. INSTALACIÓN DEL NÚCLEO PRINCIPAL (FUENTE PROPIA).....	131
FIGURA 85B. INSTALACIÓN DEL NÚCLEO PRINCIPAL Y EL SOPORTE DE LOS SENSORES DE ULTRA SONIDOS(FUENTE PROPIA). .	131
FIGURA 86. NÚCLEO PRINCIPAL JUNTO CON EL SOPORTE DE ULTRASONIDOS EN VISTA LATERAL (FUENTE PROPIA).....	132
FIGURA 87. INSTALACIÓN DEL NÚCLEO PRINCIPAL Y LA TAPA SUPERIOR CON EL ANILLO DE LEDs (COMPROBACIÓN DE ESTABILIDAD) (FUENTE PROPIA).....	132
FIGURA 88. PRESENTACIÓN SOBRE EL ROBOT F2 DE LA TAPA SIN <i>USDAR</i> (FUENTE PROPIA).....	132
FIGURA 89. PRESENTACIÓN DE LA VERSIÓN <i>USDAR</i> SOBRE EL ROBOT DE LA FASE 2 (FUENTE PROPIA).....	133
FIGURA 90. PRUEBAS DEL <i>USDAR</i> SOBRE EL ROBOT DE LA FASE 2 (FUENTE PROPIA).....	133
FIGURA 91. ESQUEMA DEL SISTEMA IMU-ENCODER (FUENTE PROPIA).....	134
FIGURA 92. ESQUEMA DEL SISTEMA IMU-ENCODER POSICIÓN 2 (FUENTE PROPIA).....	135
FIGURA 93. ESQUEMA DEL SISTEMA IMU-ENCODER SOBRE EL ROBOT (FUENTE PROPIA).....	135
FIGURA 94. VISTA SUPERIOR DEL CIRCUITO V3 (FUENTE PROPIA).....	138
FIGURA 95. VISTA INFERIOR DEL CIRCUITO V3 (FUENTE PROPIA).....	138
FIGURA 96. VISTA INTERIOR DEL ANTEBRAZO V2 (IZQUIERDA) Y REAL (DERECHA) (FUENTE PROPIA).....	139
FIGURA 97. VISTA MONOCROMA DEL ANTEBRAZO V3 (IZQUIERDA) Y SIMULADA CON EL INTERIOR VISIBLE (DERECHA) (FUENTE PROPIA).....	140
FIGURA 98. CREACIÓN DE LOS CANALES PARA LOS TORNILLOS DEL BRAZO V2 (FUENTE PROPIA).....	140
FIGURA 99. MODELO 3D DEL BRAZO V3 (FUENTE PROPIA).....	141
FIGURA 100. MODELO 3D DEL PASACABLES (FUENTE PROPIA).....	142
FIGURA 101. MODELO 3D DE LA TAPA DEL PASACABLES (FUENTE PROPIA).....	142
FIGURA 102. MODELO 3D DEL BRAZO V2 DEL PASACABLES CON SU TAPA (DERECHA) (FUENTE PROPIA).....	142
FIGURA 103. MODELO 3D DEL ROBOT EN LA FASE 3 CON <i>USDAR</i> VISTA LATERAL (FUENTE PROPIA).....	143
FIGURA 104. MODELO 3D DEL ROBOT EN LA FASE 3 CON <i>USDAR</i> VISTA FRONTAL IZQUIERDA (FUENTE PROPIA).....	144
FIGURA 105. MODELO 3D DEL ROBOT EN LA FASE 3 BÁSICA CON VISTA FRONTAL DERECHA (FUENTE PROPIA).....	144
FIGURA 106. MODELO 3D DEL ROBOT EN LA FASE 3 CON <i>USDAR</i> CON VISTA FRONTAL DERECHA DE LAS ARISTAS INTERNAS (FUENTE PROPIA).....	145
FIGURA 107. MODELO 3D DE UNA PATA DERECHA DEL ROBOT EN LA FASE 3 (FUENTE PROPIA).....	145
FIGURA 108. DIAGRAMA FINAL DE CONEXIONES (CIRCUITO_V9) DE LA FASE 3 (FUENTE PROPIA).....	147
FIGURA 109. VISTA LATERAL DEL DEFECTO DE CAPA DEL ANTEBRAZO (IZQUIERDA) Y DEL BRAZO INFERIOR (DERECHA) (FUENTE PROPIA).....	149
FIGURA 110. VISTA SUPERIOR DEL DEFECTO DE CAPA DEL ANTEBRAZO QUEBRADO A TENSION (FUENTE PROPIA).....	150
FIGURA 111. VISTA SUPERIOR DE LAS PIEZAS IMPRESAS PARA LA FASE 3 (SIN CONTAR LAS TAPAS NO VISIBLES) (FUENTE PROPIA).....	150
FIGURA 112. VISTA SUPERIOR DE LAS PIEZAS IMPRESAS PARA LA FASE 3 DEL BRAZO (FUENTE PROPIA).....	151
FIGURA 113. PREPARACIÓN PARA LA AMPLIACIÓN DEL ORIFICIO DEL BRAZO SUPERIOR (FUENTE PROPIA).....	151
FIGURA 114. TERMINAL DE CONEXIONES DE ALIMENTACIÓN SOBRE PROTOBOARD (IZQUIERDA) Y ORIGINAL DE LA FASE 1 Y 2 (DERECHA) (FUENTE PROPIA).....	152
FIGURA 115. CONEXIONES DEL ANILLO LED Y LA PANTALLA (FUENTE PROPIA).....	153
FIGURA 116. CONEXIONES DEL MULTIPLEXOR I2C (FUENTE PROPIA).....	153
FIGURA 117. NUEVAS CONEXIONES DEL LM AL <i>HUB</i> DE TENSIONES (FUENTE PROPIA).....	154



FIGURA 118. COMPARATIVA DEL NUEVO CABLEADO DE ALIMENTACIÓN A LA IZQUIERDA EL ANTIGUO Y A LA DERECHA EL MODIFICADO EN LA FASE 3 (FUENTE PROPIA).....	154
FIGURA 119. CIRCUITO TRAS LA SUSTITUCIÓN DE LOS COMPONENTES QUE PROVOCARON LOS 2 CORTOCIRCUITOS DE LA FASE 3 (FUENTE PROPIA).....	155
FIGURA 120. PRUEBA DE LA LECTURA DEL SENSOR DE PRESIÓN INSTALADO EN LA PATA (FUENTE PROPIA). ....	156
FIGURA 121. PRUEBA DE LA LECTURA DEL SENSOR DE INERCIAL MEDIANTE EL MULTIPLEXOR (FUENTE PROPIA).....	156
FIGURA 122. CÁLCULO DE LA SEÑAL PWM PARA EL CALIBRADO DE LOS SERVOMOTORES (FUENTE PROPIA). ....	157
FIGURA 123. COLOCACIÓN DEL SENSOR DE PRESIÓN (FUENTE PROPIA). ....	158
FIGURA 124. PREPARACIÓN DEL CABLEADO PARA LA SIGUIENTE ARTICULACIÓN (FUENTE PROPIA). ....	159
FIGURA 125. INSTALACIÓN Y CONEXIÓN DEL SENSOR INERCIAL (FUENTE PROPIA). ....	159
FIGURA 126. INSTANTES ANTES DE LA SOLDADURA POR FUSIÓN DEL PASACABLES (FUENTE PROPIA).....	160
FIGURA 127. VERIFICACIÓN DE LA NECESIDAD DEL PASACABLES (FUENTE PROPIA).....	160
FIGURA 128. PREPARACIÓN DEL CABLEADO PARA LA SIGUIENTE ARTICULACIÓN Y LA FIJACIÓN DE LA TAPA DEL PASADOR DE CABLES (FUENTE PROPIA). ....	161
FIGURA 129. CONSTRUCCIÓN DE LAS PATAS FINALIZADA (FUENTE PROPIA).....	161
FIGURA 130. <i>RAPID SCHEDULED DISASSEMBLY</i> DEL BRAZO PARA RESCATAR LOS COMPONENTES (FUENTE PROPIA). ....	161
FIGURA 131. PATRÓN BASE PARA EL MONTAJE DEL HOMBRO ABDUCTOR (FUENTE PROPIA). ....	162
FIGURA 132. PATRÓN BASE PARA EL MONTAJE DEL HOMBRO ABDUCTOR EN USO (FUENTE PROPIA).....	162
FIGURA 133. PIERNAS INCORPORADAS A LA MONTURA DEL HOMBRO F/E (FUENTE PROPIA).....	163
FIGURA 134. HUECOS PARA EL CABLEADO (FUENTE PROPIA). ....	163
FIGURA 135. CABLEADO INTERNO (v9_FINAL) (FUENTE PROPIA). ....	164
FIGURA 136. ROBOT TERMINADO (v3.5) A ESCALA CON UN BICHÓN MALTÉS (FUENTE PROPIA).....	164
FIGURA 137. ROBOT TERMINADO (v3.0) CON LAS EXTREMIDADES EXTENDIDAS (FUENTE PROPIA). ....	165
FIGURA 138. ELABORACIÓN DEL CONECTOR DE LA FUENTE DE ALIMENTACIÓN (FUENTE PROPIA).....	165
FIGURA 139. ELABORACIÓN DEL ALARGADOR DE LA FUENTE DE ALIMENTACIÓN (FUENTE PROPIA).....	166
FIGURA 140. PRIMERA POSICIÓN ESTABLE (FUENTE PROPIA). ....	170
FIGURA 141. ROBOT APAGADO (FUENTE PROPIA). ....	171
FIGURA 142. ROBOT EN PIE Y EN UNA POSICIÓN ESTABLE (FUENTE PROPIA).....	171
FIGURA 143. PRUEBA COMPLETA CON SENSORES DE PRESIÓN Y 1.5 KG DE AGUA (FUENTE PROPIA). ....	172
FIGURA 144. PRUEBA DE LOS SENSORES DE DISTANCIA (FUENTE PROPIA). ....	172
FIGURA 145. PRUEBA DEL SERVIDOR WEB CON IMAGEN EN DIRECTO (FUENTE PROPIA). ....	172
FIGURA 146. LECTURAS DEL IMU CENTRAL (FUENTE PROPIA).....	173
FIGURA 147. LECTURAS DEL IMU IZQUIERDO (FUENTE PROPIA).....	174
FIGURA 148. LECTURAS DEL IMU IZQUIERDO (FUENTE PROPIA).....	174
FIGURA 149A. LECTURAS DEL <i>DS4</i> EN EL NODO DE <i>ROS</i> (FUENTE PROPIA).....	176
FIGURA 149B. EJECUCIÓN CON EL GAMEPAD F310 DE <i>SPOT_MOVE</i> (FUENTE PROPIA). ....	176
FIGURA 150. MODELO DE DEL POSICIONAMIENTO DEL PIE DEL ROBOT EN LAS CURVAS DE BÉZIER (FUENTE PROPIA).....	178
FIGURA 151. ESQUEMÁTICA DE DEL POSICIONAMIENTO DEL PIE DEL ROBOT EN LAS CURVAS DE BÉZIER (FUENTE PROPIA)...	179
FIGURA 152. ESQUEMÁTICA DE DEL POSICIONAMIENTO DEL PIE DEL ROBOT EN LAS CURVAS DE BÉZIER EN MOVIMIENTO HACIA DELANTE (FUENTE PROPIA). ....	179
FIGURA 153. ALGORITMO DE APRENDIZAJE AUTOMÁTICO POR REFUERZO (FUENTE PROPIA). ....	180
FIGURA 154. ALGORITMO GENERADOR DE CURVAS PARA CADA PIERNA (FUENTE PROPIA). ....	180
FIGURA 155. POSICIÓN NEUTRAL DEL ROBOT (FUENTE PROPIA). ....	181
FIGURA 156. CINEMÁTICA INVERSA A PARTIR DE LA VARIACIÓN DEL ÁNGULO Y LA ACELERACIÓN DEL SENSOR INERCIAL Y LA PROPIA CINEMÁTICA DEL ROBOT (FUENTE PROPIA). ....	181
FIGURA 157. ENTRENAMIENTO DE LA RED POR APRENDIZAJE POR REFUERZO (FUENTE PROPIA). ....	182
FIGURA 158. EVALUACIÓN DE LA RED POR APRENDIZAJE POR REFUERZO (FUENTE PROPIA). ....	183

FIGURA 159. EVALUACIÓN MANUAL DE LA RED POR APRENDIZAJE POR REFUERZO (FUENTE PROPIA) .....	183
FIGURA 160. COMPARACIÓN DEL RENDIMIENTO EN EL APRENDIZAJE COMPARANDO EL MODELO NO ALEATORIZADO (ROJO) FRENTE AL ALEATORIZADO (VERDE) (FUENTE PROPIA).....	184
FIGURA 161. POLÍTICA DE SALIDA PARA CADA ARTICULACIÓN (FUENTE PROPIA).....	184
FIGURA 161. POLÍTICA DE SALIDA PARA CADA ARTICULACIÓN (CON ZOOM) (FUENTE PROPIA). ....	185
FIGURA 162. COMPARATIVA DE SUPERVIVENCIA ENTRE MÉTODOS (FUENTE PROPIA). ....	185
FIGURA 163. COMPARATIVA DE SUPERVIVENCIA ENTRE MÉTODOS (ZOOM) (FUENTE PROPIA). ....	186

# Índice de tablas

TABLA 1. COSTES DE IMPRESIÓN DE LA FASE 1 (FUENTE PROPIA). .....	91
TABLA 2. COSTES DE MATERIALES Y EQUIPO DE LA FASE 1 (FUENTE PROPIA). .....	92
TABLA 3. COMPARATIVA DE RESULTADOS DE LA SIMULACIÓN DINÁMICA (FUENTE PROPIA). .....	101
TABLA 4. RESULTADOS DEL ANÁLISIS DE TENSIÓN CON CARGAS DISTRIBUIDAS DE 30N (FUENTE PROPIA). .....	108
TABLA 5. RESULTADOS DEL ANÁLISIS DE TENSIÓN CON UNA ÚNICA CARGA DE 30N (FUENTE PROPIA). .....	109
TABLA 6. COSTES DE LOS MATERIALES PARA LA SEGUNDA FASE (FUENTE PROPIA). .....	125
TABLA 7. COSTES TEMPORALES Y MATERIALES DE IMPRESIÓN PARA LA SEGUNDA FASE (FUENTE PROPIA). .....	125
TABLA 8. COSTES DE TIEMPO Y MATERIALES DE IMPRESIÓN PARA LA FASE 3 (FUENTE PROPIA). .....	148
TABLA 9. COSTES DE LOS MATERIALES Y COMPONENTES PARA LA FASE 3 (FUENTE PROPIA). .....	149
TABLA 10. COSTES FINALES DEL DESARROLLO COMPLETO DEL PROYECTO (FUENTE PROPIA). .....	190

---

# Capítulo 1: Introducción

El auge de la robótica ha supuesto un cambio significativo en nuestra industria con la cuarta revolución industrial. Poco a poco lo está siendo en nuestro día a día. Desde robots que aspiran el suelo hasta robots de compañía y cuidado. Sin embargo, con el auge de las impresoras 3D y los componentes electrónicos siendo cada vez más accesibles, la robótica ha pasado de ser algo exclusivo de la industria comercial, a estar al alcance de muchos hogares y entusiastas de la tecnología con ganas de desarrollar estas tecnologías. Esto ha supuesto una aparición de nuevos robots o réplicas de modelos comerciales o de investigación, desarrollados y contruidos como hobby o proyectos personales, con unos costes muy asequibles y con comportamientos y funcionalidades casi a la altura de sus inspiraciones reales.

En este trabajo final de máster se pretende explorar las posibilidades de estos robots libres diseñados de forma particular y tratar de desarrollar sobre uno de ellos, una propuesta que trate de aunar la mayor cantidad de conocimientos adquiridos en el máster, con el objetivo de generar un prototipo funcional mejorado. Es decir, a partir de una base o modelo inicial, mejorar y modificar el diseño aplicando los conocimientos de un ingeniero con máster en automática y robótica a su posterior desarrollo, construcción e implementación.

## 1.1 Objetivos

Para lograr dicho prototipo, se deberán cumplir una serie de objetivos principales que nos permitirán desarrollar el proyecto completo. En cuanto a lo que concierne a este trabajo, plantearemos los siguientes objetivos que permitirán establecer una serie de pasos e hitos a alcanzar para obtener una primera versión funcional y trabajada.

- Planteamiento inicial del prototipo.
- Elección y estudio de una plataforma de diseño.
- Investigación de los sistemas actuales.
- Estudio de la cinemática directa e inversa del robot.
- Elección y diseño inicial de materiales y componentes necesarios.
- Modelado, rediseño y simulación del sistema.
- Construcción del robot.
- Comprobación y ajustes del sistema prototipo.
- Programación de locomoción y sensorización.

- Pruebas y evaluación de los resultados obtenidos.

## 1.2 Estructura

El trabajo está organizado en **ocho capítulos y cinco anexos**, siguiendo la normativa de estilo de la Escuela Politécnica Superior de la Universidad de Alicante. En cuanto a la distribución, se ha definido de forma lineal abordando las distintas fases, de acuerdo con la metodología de diseño de proyectos pasando de una idea básica a un prototipo.

En lo referente al contenido de los capítulos, el **segundo capítulo** abarcará el planteamiento inicial del sistema propuesto, esbozando la idea principal del mismo, así como la especificación de las metas y objetivos de los componentes que conformarán dicho prototipo. En el **tercer capítulo** se pretende explorar el estado del arte en cuanto a qué sistemas son empleados actualmente e investigar la viabilidad y aplicaciones de los distintos componentes necesarios para realizar los diseños propuestos a priori. En cuanto al **cuarto capítulo**, se pretende plantear la metodología que se seguirá en la implementación y desarrollo del sistema. Se evaluarán también las bases teóricas consultadas en el capítulo anterior y las técnicas necesarias para la implementación y creación de un prototipo completo del sistema en el **quinto capítulo**. Concluyendo con el trabajo, en el **sexto capítulo**, se comentarán y evaluarán los resultados obtenidos, brindando una serie de conclusiones y futuras mejoras para el sistema. Por último, se encuentran el **séptimo capítulo** en el que se exponen los trabajos futuros. El **octavo capítulo**, contiene la bibliografía.

En lo referente a los **anexos**, el **primer anexo** incluirá los enlaces a todos los contenidos. En el **segundo anexo** se incluyen los artículos que se han revisado durante la realización del estado del arte pero que no han sido incluidos en el **tercer capítulo**, aunque pueden resultar interesantes para trabajos relacionados y han sido útiles para el desarrollo del proyecto. En el **tercer anexo**, se encuentran los enlaces a los vídeos explicativos de la impresión 3D. En el **cuarto anexo** se hayan los enlaces a los documentos y vídeos de la simulación dinámica. Por último, en el **quinto anexo**, se encuentran los enlaces a los documentos y vídeos del análisis de tensiones.

## 1.3 Documentación

Toda la documentación del trabajo se encuentra disponible en la carpeta de google drive en [este enlace](#). En ella se encuentran: las imágenes, vídeos, códigos de prueba de los sensores, códigos finales, modelos entrenados, diagramas de los circuitos, tablas de costes, tablas de tiempos de impresión, y los ficheros STEP e IPT de los modelos 3D. Ante cualquier problema de acceso a dicho contenido, por favor, contáctame por [correo](#).

---

## Capítulo 2: Planteamiento inicial del prototipo

Debido al origen de este trabajo final de máster, hay una parte del estado del arte que fue desarrollada en un primer momento y que atañe la búsqueda de una plataforma abierta sobre la que trabajar y desarrollar. Es por esta razón por la que a continuación, se detallarán los pasos realizados para la obtención de dicho modelo. Será en el tercer capítulo, donde desarrollen los sistemas de control, modelos de investigación y comerciales actualmente disponibles, que serán necesarios para entender mejor el funcionamiento de estos sistemas, y sentar las bases para la realización del prototipo propuesto, en función de las tecnologías y métodos disponibles.

### 2.1 Búsqueda de plataformas robóticas de código abierto

En un primer momento, se plantearon dos opciones principales, poniendo de base el hecho de que se quería trabajar en un robot en el que pudiese ser empleada alguna técnica de inteligencia artificial: en primer lugar, un brazo robótico antropomórfico, de cinco o más grados de libertad, que funcionase de forma similar al robot DUM-E de la película *Iron Man*, en la que, el robot es capaz de seguir los movimientos de Tony Stark (creador del robot) y asistirle de forma inteligente basándose en comandos dados por voz. Por otro lado, un robot cuadrúpedo similar al Spot de *Boston Dynamics*, del que llevo siguiendo su evolución desde sus comienzos, y que tiene mucha autonomía. A continuación, se pueden ver en la primera Figura, ambos robots:



Figura 1. Robot DUM-E de la saga Iron Man de Marvel (izquierda)[1] y robot Spot de Boston Dynamics (derecha)[2].

A través de distintas páginas web (*Thingiverse, Google Scholar, Papers with code, GitHub, GrabCad, Thangs, Google*) y apuntes de distintas asignaturas se realizó una búsqueda inicial sobre aplicaciones desarrolladas con técnicas de impresión 3D y también aquellos dispositivos existentes en ese momento en el mercado.

Fueron muchos y muy diversos los dispositivos encontrados, pero se destacarán cinco de los candidatos más significativos. Se incluye una breve descripción cada robot, las razones por las que ese modelo fue considerado y posteriormente descartado o seleccionado. El orden visto a continuación no refleja un orden cronológico de búsqueda, sino una organización por tipos de robot, primero los antropomórficos seguidos de móviles.

En primer lugar, se destaca el robot *Moveo* de BCN3D (Figura 2)[3]. Se trata de un brazo robótico desarrollado como parte de una colaboración entre el departamento de enseñanza de la Generalitat de Cataluña y el equipo técnico de BCN3D. El brazo de seis grados de libertad cuenta con una estructura impresa mediante impresión 3D y controlada mediante el software de Arduino. El objetivo de este robot es permitir el acceso a los centros educativos a un equipamiento modificable, personalizable y con una mayor accesibilidad para el alumnado, con unos precios muy bajos en comparación con sus homólogos industriales, y cuyas capacidades y prestaciones son suficientes para suplir las necesidades académicas.



Figura 2. Robot Moveo [3].



*Moveo* cuenta con muy buenas referencias, documentación y material disponible, pero, requiere mucha inversión para su construcción, sin contar el posible rediseño de ciertos componentes para alcanzar las prestaciones que se buscan en este TFM. Además, debido a la naturaleza de la impresión 3D, no puede soportar mucho peso, ya que estos materiales son más frágiles por lo que es un robot perfecto para la enseñanza del control y manipulación, pero carece de capacidad real de profundización y versatilidad que buscaba. En definitiva, es un gran robot, pero se descartó por su precio y las pocas posibilidades de desarrollo que presenta.

El siguiente robot en el que se pensó, también antropomórfico y que sigue las líneas del *Moveo*, es el *Niryo One* de la empresa francesa *Niryo* (Figura 3)[4]. Este robot de seis grados de libertad fue desarrollado con el objetivo de crear una plataforma robótica con inteligencia artificial fácil de usar y que permita el aprendizaje de la robótica industrial. Cuenta con una estructura híbrida de materiales impresos en ácido poliláctico (PLA) y tubos huecos de aluminio que conforman el grueso de las articulaciones. Todo ello es controlado por una Raspberry y movido por una serie de servomotores, algunos de los cuales son propietarios.



Figura 3. Robot Niryo One [4].

De nuevo este robot tiene una buena documentación, plataforma desarrollada y amigable para la programación, pero con la gran mayoría de las capacidades desarrolladas, dejando solo el movimiento autónomo disponible. Al igual que el *Moveo*, su precio es bastante alto y el tamaño y capacidades lo hacen muy limitante. En definitiva, es un buen robot para el aprendizaje, con multitud de herramientas y accesorios, pero el precio de entrada al desarrollo y la falta de elementos a desarrollar lo hacen un equipo poco adecuado para un TFM.

Se comenzó una búsqueda de otros robots, no antropomórficos ya que se ha determinado que estos robots eran muy simples o con un coste de desarrollo muy elevado y en la totalidad de los casos, el aprendizaje iba a ser muy bajo. En concreto se comenzó una búsqueda de robots móviles. Estos generalmente no están limitados por la capacidad de carga y tamaño, y ofrecen mayores oportunidades. Entre los robots móviles, se prefirió el desarrollo de un robot sin ruedas dada su mayor complejidad y mayor interés.

De entre los robots móviles sin ruedas, se pensó inicialmente en el robot *Hexapod* creado por Mark W como proyecto personal. (Figura 4)[5]. Inspirado en el *PhantomX AX* de Trossen Robotics, la idea de Mark fue desarrollarlo bajando el coste y con técnicas de impresión 3D. El robot, cuenta con las mismas características que su homólogo original con un precio un orden de magnitud inferior. No solo es interesante como hexápodo, sino que es un robot que tiene muchas capacidades. Al mismo tiempo, debido a su diseño, está intrínsecamente limitado y es poco expandible, por lo que, aunque el precio es más que razonable y el robot es muy interesante, se buscaron otras opciones.



Figura 4. Robot Hexapod [5].

En una línea similar, se haya el robot *Kame* desarrollado originalmente por Javier Isabel en colaboración con el *Instituto de Innovación de Bq* (Figura 5)[6]. Este robot, de código abierto, es un cuadrúpedo de 8 grados de libertad impreso, como los anteriores, con técnicas de impresión 3D, y que emplea una ESP8266 corriendo Arduino. El proyecto no tiene mucha documentación per sé, aunque si que ofrece los modelos de las piezas. Es una apuesta interesante y tiene 3 variantes adicionales de tamaño, pero de nuevo, es un sistema limitado en cuanto al aprendizaje y desarrollo, por lo que también se descartó.



Figura 5. Robot Kame mini [6].

Finalmente, se encontró el robot *Spot micro*, originalmente creado por Deok-yeon Kim y actualmente desarrollado por toda una comunidad de *makers* con una gran cantidad de versiones (Figura 6)[2]. Diseñado a escala de su homólogo de Boston Dynamics, Spot micro, es un cuadrúpedo de 12 grados de libertad diseñado para ser impreso utilizando impresión 3D y abierto a modificaciones. La comunidad que hay detrás y la documentación disponible es extensa y variada. El robot, cuyo precio de entrada es razonable, permite una extensa modificación y ampliación de características a la vez que aprendizaje y desarrollo. Si bien es cierto que parte de los componentes hardware y software básicos del robot ya están desarrollados, la posibilidad de escoger la sensorización, equipo electrónico, modificación de componentes e implementación de comportamientos complejos, hacen ideal esta plataforma para el desarrollo de un TFM.

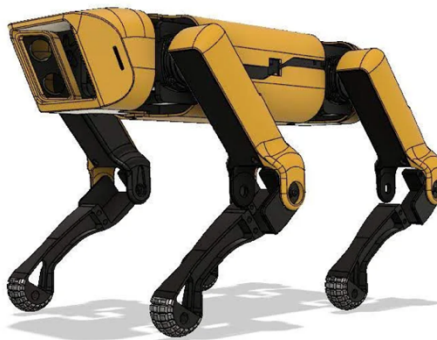


Figura 6. Robot Spot micro original [7].

Una vez seleccionado la plataforma *Spot micro*, el siguiente paso fue definir los objetivos y requisitos para el prototipo, así como elaborar una serie de pasos a seguir para desarrollarlo.

## 2.2 Elección y planteamiento inicial de la elaboración del prototipo

Tras analizar a fondo los archivos del prototipo, se decidió optar por la versión más actualizada, para así aprovechar las pruebas que ya habían realizado los miembros de la comunidad de *Spot*. Finalizada la búsqueda, se seleccionó la implementación de *Spot* disponible en el repositorio de Michael Kubina [8], que había realizado la labor de rehacer los archivos STL originales a un formato libre con *FreeCAD*, eliminando la necesidad de soportes en la impresión. Esto último se consideró importante pues reducía el material de impresión desperdiciado y no incrementaba la complejidad del proceso de preparación de las piezas. Por último, hay que destacar que también emplea como punto de partida la plataforma ESP32, que se consideró ideal para controlar el robot.

Una vez seleccionado el robot, se contactó con el desarrollador y se comenzó una colaboración para el desarrollo de “*remix*” del robot, aportando conocimientos en la materia y variaciones. Se acordó con el desarrollador que, salvo por cuestiones de electrónica, parte del desarrollo que se realizaría sería público una vez presentado este TFM.

Teniendo decidido el diseño y modelo, se establecieron los pasos a seguir para desarrollar el prototipo, que serían:

1. Definición y análisis de capacidades, características y objetivos.
2. Estudio de los requisitos para llevar a cabo el prototipo.
3. Estudio del estado del arte de robot cuadrúpedos.
4. Estudio y selección de componentes.
5. Elaboración de diagramas de coste y tiempos de producción.
6. Construcción de una primera versión simple.
7. Desarrollo de una segunda versión incorporando mejoras y corrigiendo errores básicos.
8. Desarrollo de una tercera versión que cumpla con los objetivos.
9. Comprobación de sistemas.
10. Programación del control.
11. Pruebas.
12. Estudio de trabajos futuros

De esta forma, se estableció una metodología de aprendizaje y mejora gradual que me permitiese mejorar el prototipo gradualmente en base a la teoría y la experiencia. Esto nos permite obtener un mejor robot final, en comparación con una implementación inicial a ciegas, pues nos permite analizar mejor los límites y capacidades de la máquina sobre la que trabajaremos.

## **2.3 Análisis de capacidades, características y planteamiento de los objetivos a realizar**

Una vez establecido el plan, era necesario evaluar las capacidades a priori del robot y aquellas características que lo definen. Para ello, se detallará una lista de los sensores y componentes necesarios para que el robot funcione y para los que hay un alojamiento en el chasis:

- Dos sensores frontales de tiempo de vuelo.
- Una ranura para cámara.
- Doce servomotores.
- Batería.
- Un transformador.
- Placa de control.
- Sensor inercial.
- Relé.
- Pulsador.
- Placa de control de servomotores.

Con este conjunto de sensores y equipamiento, se puede lograr un movimiento completo, sin mucho control, evitando obstáculos frontales. Las piernas del robot están diseñadas de forma estilizada y minimizando el peso. En el interior del chasis, las proporciones implican que hay un espacio compacto poco ampliable y que, por tanto, hay una limitación en la cantidad de sensores que podemos añadir y en la autonomía.

Conociendo mejor sus características, se pudieron establecer unos objetivos deseables para que cumpliese el robot. Se parte de estos como base para realizar el estudio de los requisitos necesarios y sistemas a desarrollar a lo largo de este proyecto con el fin de obtener la mejor versión posible dadas las limitaciones. Estos objetivos iniciales son:

- Información visual del estado del robot.
- Control preciso de las articulaciones.
- Control del estado y autonomía de la batería.
- Movimiento natural/inteligente.
- Simulación del robot.
- Mejora de la sensorización.
- Monitorización externa.

## **2.4 Análisis de los requisitos para llevar a cabo el prototipo**

Definidos aquellos objetivos iniciales, se procedió a elaborar los requisitos de desarrollo e investigación para cada uno de ellos. El propósito de esta sección es evaluar qué conocimientos o desarrollos deberían ser realizados para lograrlos. De esta forma, pasamos a definir y focalizar los esfuerzos de manera concreta, guiada y estructurada.

### **2.4.1 Información visual del estado del robot**

El robot no contaba en un principio con ningún tipo de sistema visual para adquirir información de su estado; solo contaba con un par de LEDs en la base que eran meramente estéticos. Por tanto, se consideró necesario inicialmente desarrollar las siguientes características y aptitudes:

- Brindar información acerca del estado actual de la batería.
- Ofrecer información del modo en el que se encuentra.
- Mostrar visualmente los obstáculos encontrados.
- Mostrar el estado de la conexión inalámbrica.

Para ello es necesario estudiar la posibilidad de integrar un conjunto de LEDs y/o una pequeña pantalla que pueda mostrar esta información al usuario de forma sencilla.

### **2.4.2 Control preciso de las articulaciones**

El robot inicialmente se controlaba sin emplear ningún tipo de sistema inteligente ni algoritmo de control real. Esto se debe a que, a pesar de contar con un sensor inercial, la información del posicionamiento de las articulaciones era

desconocido y se basaba principalmente en un “control” en bucle abierto, en el que se confiaba que los controladores internos de los servomotores estuviesen en la posición indicada y con el terreno llano. Por tanto, al no contar con los sensores suficientes, ni con encoders para implementar ningún tipo de control convencional, era necesario:

- Expandir el número y el tipo de sensores para lograr un mayor control.
- Obtener mayor precisión y control de las articulaciones.
- Implementar un sistema de control más avanzado.
- Evaluar los motores propuestos y alternativas posibles.

Para lograr todo esto fue necesario realizar un estudio de los sistemas de control empleados en la actualidad, evaluar y estudiar las necesidades de estos para extraer posibles sensores. De esta forma se pretende lograr ese mayor control y precisión sobre las articulaciones y calcular qué tipo o modelo de motor es el más apropiado para estas tareas.

#### 2.4.3 Control del estado y autonomía de la batería

Originalmente y, siguiendo la simplicidad en el diseño, el robot contaba con una batería que se conectaba directamente al regulador de potencia. Esta implementación, aunque funcional, brinda menos control sobre el sistema. Este control es necesario, pues los servomotores y componentes tienen unas intensidades máximas soportadas antes de sufrir daños que pueden comprometer a todo el sistema. Por tanto, era necesario:

- Incorporar la sensorización apropiada para el control de la tensión y la intensidad en el circuito.
- Estudiar los consumos para la elección correcta de la batería.
- Estudiar la compatibilidad de dichos sensores con un sistema de muestreo de información del punto 2.4.1.
- Explorar fuentes de alimentación alternativas para las pruebas.

Para lograr estos requisitos, se buscarán dichos sensores y cómo pueden ser incorporados al sistema, y se evaluarán los consumos globales para escoger la mejor batería para el sistema, se estudiarán alternativas de alimentación para realizar pruebas de larga duración sin desgastar la batería y se complementará con lo desarrollado para el punto 2.4.1.

#### 2.4.4 Movimiento natural/inteligente

El robot inicialmente no contaba con un sistema de control capaz de operación lejos de unas condiciones ideales y muy controladas. En parte, este problema surge de la implementación de motores y del algoritmo de control mencionado en el apartado 2.4.2. El objetivo de la implementación era que fuese capaz de operar independientemente del terreno en la medida de lo posible con el foco en un movimiento lo más natural posible. Para lograr esto, es necesario:

- Sensorización que brinde conocimientos del entorno y del movimiento.
- Articulaciones ágiles, precisas y con un movimiento “suave”.
- Elección del lenguaje y plataforma de control.
- Un control más robusto.
- Capacidad de movimiento semiautónomo.

Por tanto, en líneas generales, se consideró necesario desarrollar un control preciso de las articulaciones (punto 2.4.2), así como estudiar la incorporación de un algoritmo o método de control adicional que se encargue de gestionar la navegación en terrenos irregulares de manera independiente a la orden de movimiento. Finalmente, también se consideró estudiar qué plataforma de control y lenguaje es el más apropiado para la realización de las distintas funciones previstas.

#### 2.4.5 Simulación del robot

Con el objetivo de poder probar los sistemas de control y el código generado sin dañar el prototipo, fue necesario realizar un modelo para la simulación. Para lograrlo, se necesita:

- El modelo ensamblado en CAD/CAE.
- La creación de un modelo completo en URDF.
- Definición del modelo en Gazebo/RVIZ con los sensores adecuados.

#### 2.4.6 Mejora de la sensorización

Este objetivo surge de la necesidad de conocer mejor el entorno y no sólo lo que el robot tiene en frente. Con esa finalidad en mente, se exploraron las siguientes opciones:



- Implementación de un sistema igual o similar a un LiDAR (Light Detection and Ranging o Laser Imaging Detection and Ranging).
- Incorporación de sensores de tiempo de vuelo por ultrasonidos en más puntos.
- Incorporación de sensores de infra rojos (IR).
- Dotación de un algoritmo inteligente a la cámara frontal.

Para todo ello, era necesario comparar los distintos sensores en los casos prácticos correspondientes para evaluar mejor su utilidad, investigar técnicas de reconocimiento de superficies y patrones empleando visión artificial y, por último, plantear o estudiar la incorporación de un sistema de detección a distancia.

#### 2.4.7 Monitorización externa

Por último, surge la necesidad de poder monitorizar de forma externa al robot, para aquellos casos en los que esté siendo tele operado, así como en los casos en los que pueda estar bajo un control completamente autónomo y fuera de nuestro campo visual. Para ello se establecieron los siguientes requisitos:

- Acceso remoto a la cámara del robot.
- Acceso a los datos de telemetría del robot.
- Posibilidad de control externo.
- Simulación del movimiento en tiempo real.

Para ello fue necesario investigar los métodos y protocolos de comunicación que permiten esta monitorización externa, la posibilidad de alimentar el modelo simulado con la telemetría del robot real y explorar los métodos de teleoperación disponibles.

---

## Capítulo 3: Estado del arte

En este tercer capítulo, una vez explorados todos los objetivos y requisitos, y tras establecer las vías de desarrollo necesarias para llevar a cabo el prototipo, se procedió a realizar la investigación del estado del arte para situar el proyecto en contexto, recopilar toda la información necesaria para continuar con el desarrollo, así como poder evaluar la viabilidad de los objetivos propuestos y sus implementaciones en el mercado. En el siguiente apartado se incluyen algunos de los muchos robots con locomoción desarrollados a lo largo de la historia excluyendo las plataformas actuales, que serán mencionadas en el apartado 3.2.

### 3.1. Breve historia e introducción a la robótica móvil cuadrúpeda

Desde finales del siglo pasado la robótica móvil ha sido el centro de atención debido a su capacidad de exploración en entornos complejos, en operaciones de salvamento, la exploración espacial, tareas autónomas, etc. El robot móvil puede clasificarse de manera simplificada en las siguientes tres categorías: robot con orugas, con patas y con ruedas. El sistema de locomoción de un robot es una característica esencial para el diseño de la robótica móvil. No solo depende del espacio de trabajo que posea, sino también de una serie de mediciones técnicas como la maniobrabilidad, la capacidad de control, la eficiencia, la estabilidad del sistema y las condiciones en las que se encuentra el terreno a navegar por el robot [9][10]. Cada uno de estos tipos de sistemas móviles, presenta una serie de ventajas y desventajas frente al resto. Algunos autores [9] han hecho comparativa detallada.

El desarrollo de los sistemas de locomoción terrestre para el robot con patas/extremidades ha ido en aumento a lo largo de estas últimas décadas debido a su mayor número de ventajas con respecto a los robots con ruedas. Aún así, dichas ventajas que poseen los sistemas con locomoción de piernas son dependientes del número de piernas, sus funcionalidades y las posturas que adoptan o pueden adoptar, presentando en general un mayor potencial para navegar por casi todas las superficies, al igual que lo hace el ser humano y los animales [9][10].

Entrando concretamente en la historia de los robots cuadrúpedos, podemos ver los primeros intentos de lograr sistemas básicos de locomoción a finales del siglo XIX. Estos no eran robots, tal y como los conocemos hoy en día y nacen décadas antes de la aparición de la propia palabra. Sin embargo, estos aparatos dieron pie al desarrollo de la robótica móvil que conocemos hoy en día. De esta época, se destaca la invención por parte del matemático Pafnuty Chebyshev que porta su apellido, del primer mecanismo de locomoción autónoma con cuatro extremidades en el año 1870. Este dispositivo, que solo era capaz de caminar de forma dinámica en terreno plano, no contaba con un movimiento de piernas independientes. Posteriormente, este mecanismo se incorporó a dos máquinas Melwalk y Dante [11]. También de finales del diecinueve, y quizás más conocido en la literatura americana, surgió el caballo mecánico patentado por L.A. Rygg en 1893.

Unas cuantas décadas después, en la década de los sesenta, observamos la aparición del primer robot cuadrúpedo autónomo en los EE. UU. Este robot, *Phoney Pony* [9][11], fue construido en la Universidad del sur de California. Cada una de las extremidades de este robot contaba con dos articulaciones rotativas idénticas accionadas eléctricamente y capaces de producir varios patrones de marcha como caminar, gateo a baja velocidad y trote. Además de estas funciones, gracias a que cada pie estaba inspirado en forma de T invertida o estructura pélvica, era capaz de proporcionar la estabilidad en el plano frontal lo que le permitía mantener una marcha erguida estable y estacionaria. El control del robot se realizaba mediante el uso de biestables y lógica eléctrica, mediante la cual el cuadrúpedo realizaba el rastreo y el trote diagonal en función de la marcha programada en el diagrama de estado seleccionado en el momento.

Algo más adelante aparecen los ya famosos robots humanoides de HONDA, que, aunque se centraron en la locomoción bípeda con sus series E, P y el posterior ASIMO [12], han servido de inspiración para muchas plataformas tanto cuadrúpedas como humanoides así como también han promovido la robótica móvil en general.

En un punto intermedio entre las series E y P de honda, aparece la serie Titan (III a XIII) de Shigeo Hirose [13], a finales de los años ochenta hasta su última versión lanzada en 2008. Inspirado en el robot PV-II [9], se trata de una línea de robots cuadrúpedos con la habilidad de navegar por terrenos con pendiente y portar cargamento y sensores (últimas versiones). Los robots Titan han participado en labores de asistencia en la construcción incluso por andamios

en pendiente, labores de ayuda humanitaria, perforación y taladro de superficies en pendiente. Todo esto en el caso de la versión XII a una velocidad máxima de 1,5m/s y en la última versión, la XIII, a 1,38 m/s alimentado con baterías.

Por último, antes de pasar a las plataformas cuadrúpedas actuales, hay que destacar la serie TEKKEN desarrollada por H. Kimura y su equipo. [9][14], que logró diferentes modos de locomoción dinámica: trote, galope con un solo mecanismo, caminar y un método de control inalterable basado en un sistema neuronal y generador de patrones central (CPG).

### **3.2. Plataformas robóticas cuadrúpedas actuales**

Explorada la historia de la robótica móvil cuadrúpeda, se realizó un estudio de los sistemas más actuales de los que podríamos extraer información o ideas aplicables a la plataforma que se pretende desarrollar. A continuación, se hablará de algunos de los robots destacados, explorando sus capacidades, similitudes e innovaciones aportadas a la industria.

En primer lugar, se encuentran los robots cuadrúpedos de Boston Dynamics, que han servido para llegar al robot *Spot*, sobre el que se basa el prototipo a desarrollar. Se distinguen 2 ramas principales desde su concepción (militar y civil multipropósito), alineándose con el cambio de intereses de la empresa. Comenzaron en 2004 con su robot BIGDOG (Figura 7a), este fue el primer robot con patas que crearon, siendo capaz de navegar por terrenos accidentados, empleando distintos sensores y un novedoso sistema de control. Unos años más tarde, en 2010, fue el turno de LS3 (Figura 7b), diseñado desde un principio y en base a la plataforma anterior, para seguir a los soldados y llevar su equipo por terrenos accidentados.

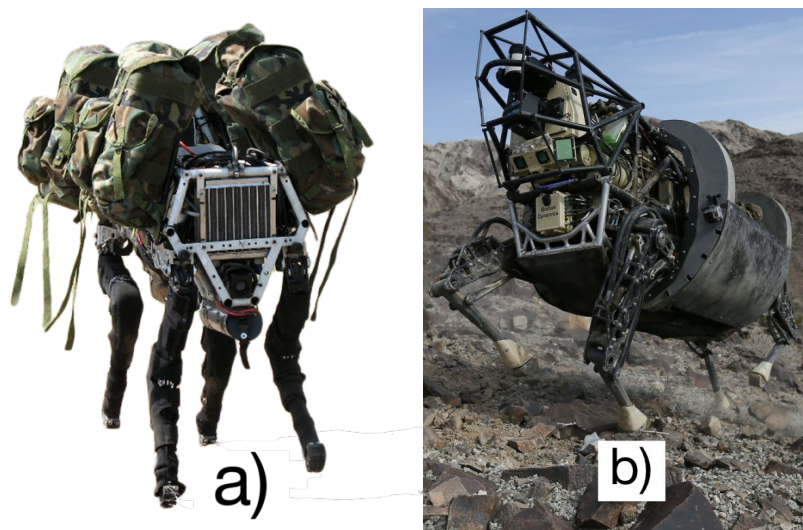


Figura 7. Robots BIGDOG (a) y LS3 (b) de Boston Dynamics [15].

Cinco años después, tras trabajar en múltiples robots caminantes, entre ellos el bípedo *Atlas* (bípedo más avanzado), y el cuadrúpedo *Wildcat* (con el actual récord de velocidad en cuadrúpedos con una velocidad máxima de 32 km/h), desarrollaron una versión más compacta cuadrúpeda llamada *Spot* (ahora *Spot classic*) (Figura 8a). *Spot* fue concebido en 2015 como un robot de estancia híbrida (interiores y exteriores) cuyo propósito fue ser una plataforma de desarrollo para una versión más avanzada con un robusto control dinámico que incorpora la actual iteración de *Spot*, desarrollado como *Spot Mini* y lanzado al mercado comercial a finales de 2019 como *Spot* (Figura 8b). *Spot* está diseñado para ser ampliable y escalable y así cumplir con distintas aplicaciones.

Se trata de un cuadrúpedo de 12 grados de libertad con motores y encoders independientes, y está completamente sensorizado con cámaras de profundidad, dotándole de 360° de visión espacial y autonomía para la navegación (los sensores internos se desconocen, pero se espera tenga sensores inerciales, de presión, temperatura...). Su sistema de control se basa en algoritmos de control inteligentes haciendo uso de redes neuronales propietarias que, junto con el resto de los sensores internos, logran la versatilidad del robot. En otros aspectos, es capaz de cargar hasta 14kg, integrar sensores, equipo e incluso un brazo robótico. Es resistente al polvo y al agua, permite conectividad vía wifi y es completamente autónomo. Por último, cabe destacar el hecho de que cuenta con distintos modos: manual tele-operado, semiautónomo (por tareas autónomas bajo control manual) y modo completamente autónomo.



Figura 8. Robots Spot classic (a) y Spot (b) de Boston Dynamics [2][15].

Siguiendo la línea de investigación cuadrúpeda se encuentran los robots *Chetaah* 1 a 3 [16] (Figura 9a), *Cheetah-cub-S* [17] y el *Minichetaah* (Figura 9b) [18], del Instituto Tecnológico de Michigan (MIT) en los EEUU. La serie *Cheetah* sigue unos pasos similares a Boston Dynamics en cuanto al desarrollo de una plataforma cuadrúpeda autónoma multipropósito y, en cuanto a su diseño, comparten muchas similitudes. No obstante, hay una distinción importante y es que estos robots son mucho más acrobáticos que sus homólogos bostonianos.

La serie *Chetaah* se centra en la agilidad y la versatilidad. Y es que, incluso contando con un equipo de sensores muy similares a los ya mencionados en la plataforma *Spot*, el robot *Cheetah* 3 no solo es capaz de subir y bajar por las escaleras sin usar la visión artificial si no que también realiza saltos mayores, galopa a través de superficies con desniveles y es capaz de una recuperación rápida del equilibrio al ser molestado, función que posee *Spot* también, pero sin necesidad de valerse de sus cámaras. Esto de nuevo es posible gracias al uso de sofisticados sistemas de control y de las redes neuronales que operan el sistema. En general son robots con características similares a *Spot*, incluso la versión más compacta y perfilada, *Minichetaah*, más similar en acabados, pero no han visto aún un despliegue comercial más allá de pequeñas demostraciones aisladas.

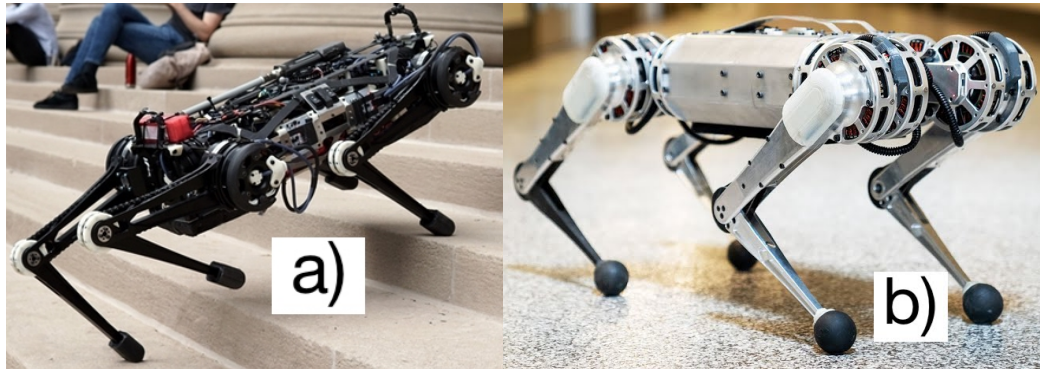


Figura 9. Robots Chetaah 3 (a) y MiniChetaah (b) del MIT [16][18].

De nuevo en una línea similar se encuentra la empresa china, UniTree Robotics [19], con sus robots (Figura 10) *A1*, *Aliengo*, *Laikagoo*, *Go1* y *Ben Ben*, visibles en la siguiente imagen compuesta:

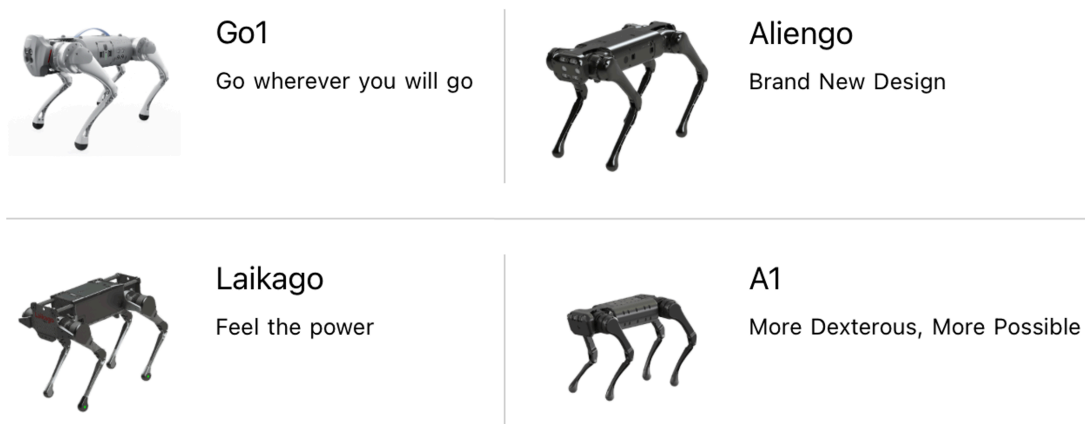


Figura 10. Robots de UniTree Robotics [19].

De nuevo se observa una línea similar en cuanto al diseño y una apuesta por una sensorización basada en cámaras de profundidad o puramente estéreo, que, junto con un equipo de sensores internos y la ayuda de algoritmos de control inteligente, dotan a estos robots de movimientos complejos y reactivos con el entorno. Su último modelo, *Go1*, se presenta como una alternativa económica a *Spot* estando una orden de magnitud por debajo. La documentación y especificaciones técnicas proporcionadas son bastante exhaustivas en cuanto a los motores empleados y capacidades destacando el trabajo realizado en estos últimos. En general, ofrecen unas prestaciones muy similares y refuerzan la selección de ese diseño, sensores y arquitecturas de control.

Finalizando con grandes líneas de desarrollo se encuentra *Ghost Robotics* [20] con sede en los EEUU. La empresa cuenta con una plataforma robótica (Figura 11) con 3 variantes principales y 20 especializaciones centrada principalmente en contratos de defensa. En cuanto a las capacidades, aparentemente deberían de ser similares a las ya descritas, pero ofrecen poca información al respecto en el momento de la investigación del estado del arte. Sin embargo, tienen un robot funcional y prometen dar buenos resultados. Una de sus versiones se encuentra en la siguiente imagen:



Figura 11. Solución Zero Point de Ghost Robotics [20].

Finalizando este capítulo, se explorarán algunas plataformas robóticas que son menos conocidas o con una única implementación, de forma más esquemática incluyendo robots impresos con impresora 3D, con propulsión neumática, comerciales y en desarrollo.

- **Robot trepador bioinspirado de la Universidad Metropolitana de Tokyo:** este robot cuadrúpedo de 23 GdL (Figura 12), es capaz de subir escaleras, está impreso con técnicas 3D y cuenta con una unidad de medición inercial (IMU), una cámara 3D de tiempo de vuelo, sensores de tacto y de fuerza en cada garra. Su unidad de control está compuesta por un Intel NUC y un Arduino como controlador secundario para administrar las señales de entrada y salida de los sensores [21] .



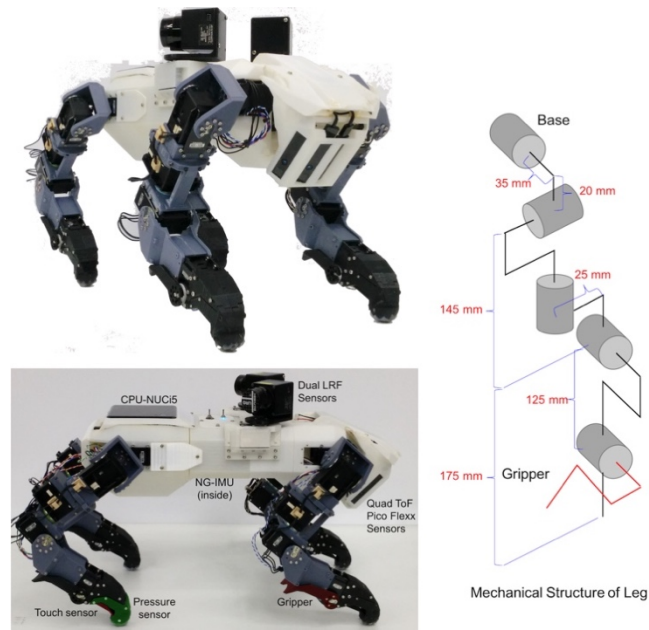


Figura 12. Robot bioinspirado de la TMU[21].

- **Robot de código abierto cuadrúpedo:** con una base similar a un *Cheetah*, este cuadrúpedo DIY [22], permite realizar funcionalidades similares a *Spot Micro* montando una raspberry Pi3 como controladora y una Arduino mega como interfaz de conexiones. El diseño no está refinado, pero ofrece una plataforma de desarrollo y pruebas.
- **Robot HYQ del Instituto Italiano de Tecnología:** lanzado en 2012 (Figura 13), es capaz de moverse a velocidades de 2m/s realizando trote [23] [24]. Forma parte de la línea de investigación robótica y es capaz de adaptarse a terrenos irregulares e impredecibles. Cuenta con sensores inerciales y contaba con planes de incorporarle dos manipuladores robóticos frontales.

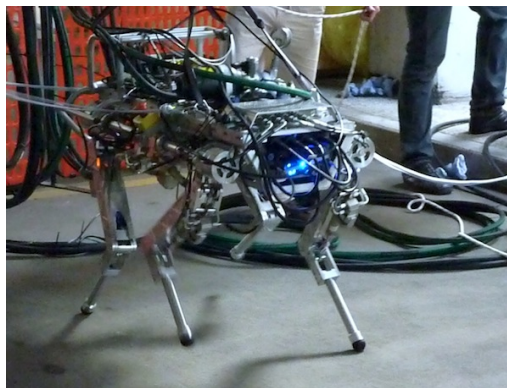


Figura 13. Robot HYQ del Instituto Italiano de Tecnología [24].

- **Robot Rush:** desarrollado por Z.G. Zhang y H. Kimura en 2008 como parte de una colaboración entre Tokyo University of Science y Kyoto Institute of Technology, para estudiar el funcionamiento autónomo y eficiente en terrenos planos y accidentados con robots cuadrúpedos. Se trata de un sistema compacto (Figura 14), con un solo actuador por pata, lo que difería de los sistemas desarrollados hasta entonces y le proporcionaba una estrategia de control novedosa basada en carrera energéticamente eficiente [25].

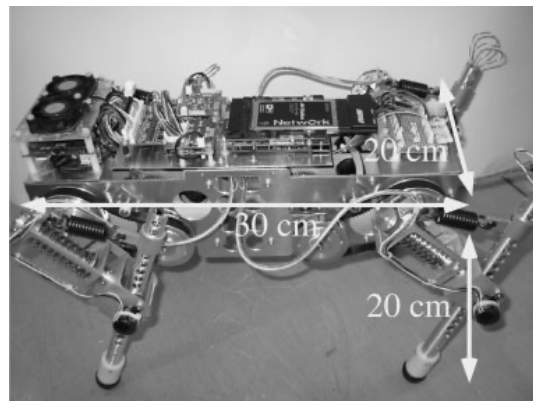


Figura 14. Robot Rush [25].

- **ANYmal de ANYbotics AG:** se trata de un robot cuadrúpedo robusto (Figura 15) con enfoque industrial y con prestaciones de autonomía y navegación como los sistemas anteriores [26]. Sin embargo, al ser un sistema más grande sufre de menor velocidad de movimiento que compensa con excelentes prestaciones de agilidad con los obstáculos. La autonomía es baja y se basa principalmente en tecnologías de cámaras y LiDAR junto al resto de sensores para inspecciones industriales.



Figura 15. Robot ANYmal de ANYbotics AG [26].

- **Pneupard de la Universidad de Osaka:** se trata de una plataforma bioinspirada de 4.8 kg (Figura 16), menos desarrollada que los anteriores, y que experimenta utilizando músculos neumáticos como motores para el movimiento imitando a un perro. Este tipo de sistemas actuadores también es conocido como mecanismos complacientes, en cuanto a que son más suaves al movimiento y no requieren frenar de golpe como en un motor eléctrico al parar el movimiento, permitiendo movimientos más continuos y eficientes. Fue creada en 2014 [27] basándose en desarrollos anteriores como *Puppy* de 2006 [28] y logró que el robot caminase sosteniendo su peso y empleando un sistema de control con realimentación en lazo abierto.

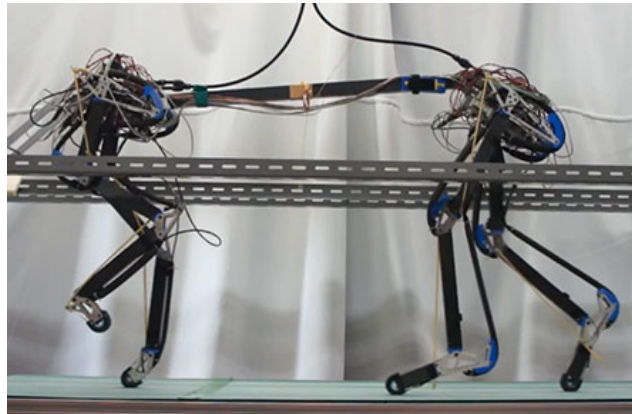


Figura16. Robot Pneupard de la Universidad de Osaka [27];

Una vez vistos los sistemas actuales y observando que la diferenciación principal reside en los sistemas de control de estos robots, se procedió a indagar en el estado del arte de los sistemas de control.

### 3.3. Sistemas de control de robots

Existen multitud de enfoques y sistemas que se han desarrollado o están desarrollando para aplicaciones de locomoción en robots cuadrúpedos. A priori se realizó un estudio recogiendo una serie de artículos de investigación con el objetivo de evaluar la viabilidad de estos métodos, aplicados al prototipo a desarrollar. Sin embargo, tras focalizar el alcance de este trabajo y sus objetivos, se decidió centrarse para esta parte del estado del arte, en aquellos sistemas que se estaban implementando en la plataforma *Spot micro* o similares, pues es sobre la que se va a trabajar. De este modo, podemos evaluar mejor qué se

puede aportar y que limitaciones se han hallado las personas que ya han desarrollado o trabajado con el robot. No obstante, en el segundo anexo, se pueden encontrar referencias a estos artículos y una breve descripción de su aportación al campo de la robótica móvil cuadrúpeda.

### 3.3.1 Control directo sin controlador

Este tipo de métodos de control funcionan aplicando un conocimiento básico de la cinemática del robot, a menudo sin tenerla en cuenta, para realizar movimientos complejos. Este tipo de movimientos han sido a menudo realizados a mano, calibrando cada paso y dependen de unas condiciones muy precisas en el entorno para operar.

En cuanto al movimiento, este se logra generando conjuntos de movimientos para cada una de las articulaciones y coordinándolas para avanzar. En todos los casos, un dispositivo de control (un mando o teclado) es necesario para la operabilidad del robot (3.6.1). Algunos de estos sistemas que incorporan algún tipo de control inteligente de la estabilidad empleando un sensor inercial (3.4.4), que ajusta la posición de los servomotores en función de la inclinación del robot [29]. También hay implementaciones con evasión de obstáculos básica (modificación de la trayectoria) empleando sensores de ultrasonidos (3.4.1) y movimiento diferencial o giro [30]. Existen muchas más implementaciones similares disponibles de forma visual en la plataforma YouTube bajo la búsqueda de *SpotMicro movement*.

### 3.3.2 Control con trayectorias generadas automáticamente sin controlador

A pesar de que el nombre de este sistema pueda incitar a una contradicción por la mención a la generación automática de trayectorias, el sistema no emplea ningún algoritmo de control cinemático convencional. El funcionamiento del sistema consiste en la introducción de los datos del robot (posiciones de los servomotores) de forma similar a la cinemática del robot, pero sin necesitar el modelo completo del robot. Seguidamente se introducen los valores de calibración de los servomotores y se programa en C qué acciones queremos que hagan cada una de las articulaciones.

El programa, *Coreograph*, permite guardar conjuntos de movimientos (*snapshots*) para crear coreografías de movimientos más realistas a partir de estos movimientos sencillos. El resultado es la ilusión de comportamientos complejos que de nuevo dependen de unas condiciones precisas para funcionar, y son llevadas a cabo al asignarse a botones en un mando controlador. Este sistema esta disponible como un proyecto en *GitHub* para usarse en *SpotMicro* y cuenta con soporte en Francés e Inglés [31]. Por último, este tipo de implementaciones se llevan a cabo sobre una placa tipo Arduino con un módulo *Bluetooth* y en algunos casos, con una *Raspberry Pi*.

### 3.3.3 Control directo con cinemática inversa

Este tipo de sistemas emplean controladores PID para el control de los servomotores junto con un modelo de la cinemática inversa para el cálculo del posicionamiento de las articulaciones a lo largo de una trayectoria. El funcionamiento de este tipo de implementaciones sigue el siguiente esquema de control en lazo abierto, ya que no hay información precisa de los motores realimentada al controlador, aunque puede que haya de otros tipos (en rojo):

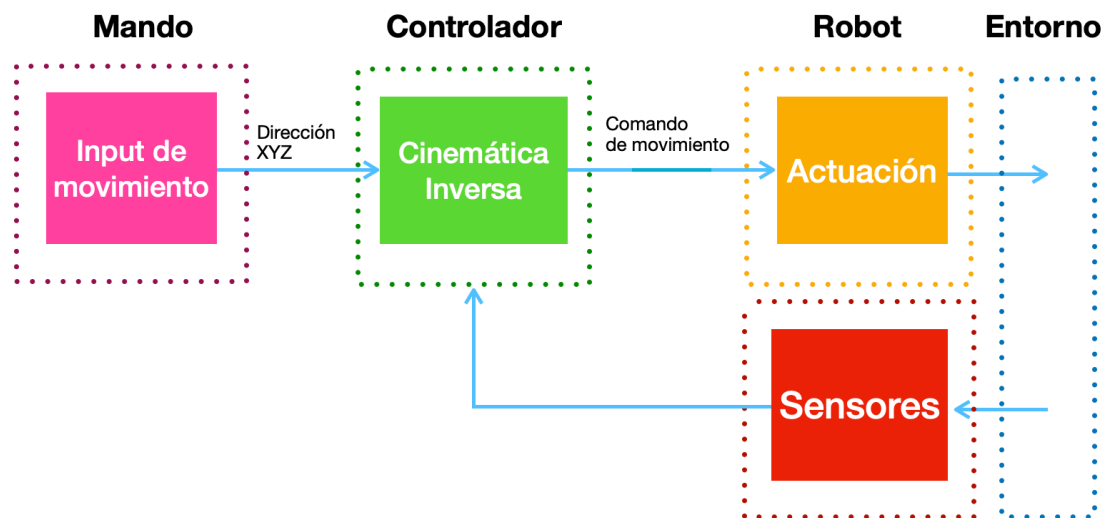


Figura 17. Esquema de control con cinemática inversa (Fuente propia).

Es el segundo tipo de implementación más común y el que permite realizar un control básico del robot. En general, no suelen emplear simulaciones y se acompañan de un mando remoto como se explica en el punto 3.6.2, para enviar comandos direccionales, que el sistema traducirá en trayectorias y posiciones. Aunque existen varios repositorios que implementan este tipo de sistemas,

muchos mencionados en los siguientes apartados, aquí se pueden encontrar los que presentan resultados satisfactorios. El primero forma parte de la comunidad *SpotMicro* y cuenta con gran documentación y guía para una implementación básica con *Arduino* y *RaspberryPi* [32] y el segundo, aunque menos documentado, implementa un simulador con *Pygames* adicional [33].

### 3.3.4 Control con curvas de Bézier y aprendizaje automático por refuerzo (RL)

Este método de control de la marcha en lazo abierto (*Open-Gait Control*) esta basada en la implementación inicial del robot *Chetaah* publicada en el año 2014 por los investigadores del MIT. En ella se plantea el uso de estas curvas para el movimiento óptimo al caminar por parte de su robot cuadrúpedo. En nuestro caso, para una plataforma derivada de *Spot Micro*, *Open Quadruped*, existe una implementación que incorpora este sistema (Figura 18) [34]. En ella se aplican unas curvas de Bézier de doce puntos inspiradas en las bidimensionales del *Chetaah*, pero modificadas para permitir cualquier combinación de comandos de movimiento y velocidades de paso definidas por el usuario. Estas curvas proporcionadas por un generador de trayectorias se apoyan en la cinemática inversa y en un sensor inercial 3.4.4, para establecer cual es el mejor movimiento para el robot. Esto es explicado en detalle en el artículo del MIT [35]. Este tipo de control fue desarrollado para *Spot Micro* con una placa de bajo nivel tipo *Arduino* y una *Raspberry Pi* para el control a nivel superior. El trabajo se encuentra publicado en un repositorio de *GitHub* [36].

Ampliando sobre el trabajo realizado para la implementación de este sistema, los mismos desarrolladores crearon un modelo en *Pybullet* (3.5.6) derivado del modelo de *Gym* para entrenar la marcha en lazo abierto (Figura 18) con aprendizaje automático. El objetivo era conseguir no solo que el robot avanzase de forma autónoma en función de las entradas de sus sensores, sino que fuese capaz de, ante cualquier imprevisto en el terreno, adaptarse y seguir la marcha.

El sistema completo, implementado para su versión del robot emplea: un sensor inercial, cuatro sensores de fuerza opcionales y el mismo sistema de controlador que el visto anteriormente. Este desarrollo así como sus resultados están documentado en el artículo disponible en la página y en su repositorio [36]. Una ilustración del entorno de simulación para el aprendizaje es el siguiente:



Figura 18. Entrenamiento por refuerzo del robot Open Quadraped [36].

### 3.3.5 Control con navegación autónoma (SLAM)

Estas dos últimas implementaciones incorporan al modelo original de *Spot Micro* un *LiDAR* (3.4.1). Su objetivo es implementar una navegación autónoma más eficaz que con los dos sensores de ultrasonidos frontales que llevan los modelos estándar. Ambas se basan en una simulación para probar los sensores y posteriormente reconstruir el entorno, pero en su núcleo, persiguen dos objetivos diferentes:

#### 3.3.5.1 Implementación con RViz [37]

En esta implementación, se pretende mapear estancias para navegar en ellas de forma autónoma usando un modelo similar al descrito en el apartado 3.3.3, con una capa de *RViz* en *ROS* que gestione la trayectoria y evasión de obstáculos. En este caso incorpora un sistema de gestión de caminado de 8 fases y otro para trote con 4 fases siendo estas últimas, producto de la sincronización diagonal de las patas.

El entorno de simulación y programación se basa en *ROS Kinetic* y emplea una *RaspberryPi 3B* como procesador. Es un proyecto que está documentado correctamente y permite un uso algo más avanzado, aunque con un coste bastante mayor debido a los componentes utilizados. El resultado se puede apreciar en la Figura 19 visible a continuación:



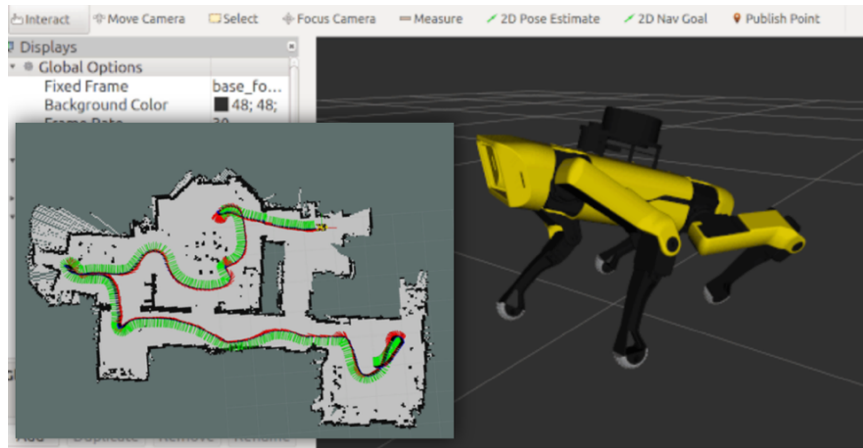


Figura 19. Simulación de Spot Micro con un LiDAR en RViz (compuesto) [38].

### 3.3.5.2 Implementación con Pybullet y ML (Machine Learning) [38]

Esta implementación no ha sido tan desarrollada como la anterior, pero implementa un enfoque distinto. En este caso el desarrollo se ha centrado en elaborar un modelo en *Pybullet*, como el desarrollado por el equipo mencionado en el 3.3.4. El objetivo del proyecto es incorporar el *LiDAR* al aprendizaje automático para lograr una autonomía de movimiento. Adicionalmente, el autor del repositorio ha desarrollado su propio modelo *URDF* de Spot Micro, así como la cinemática inversa. Esto se debe a una variación en el diseño principal. En cuanto al método de aprendizaje automático por refuerzo que se está desarrollando, el objetivo es basarse en el artículo publicado por Google en 2018 acerca del desarrollo de técnicas de locomoción ágil para robots cuadrúpedos [39]. El desarrollo promete buenos resultados. En la siguiente imagen (Figura 20) se ilustra el estado actual de simulación:

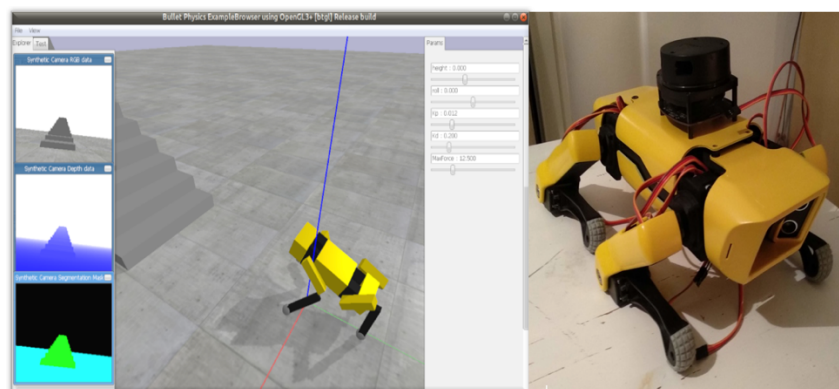


Figura 20. Simulación en Pybullet del proyecto y robot real [37].



### 3.4. Sensores y actuadores en robots cuadrúpedos

En esta sección del estado del arte se pretendía investigar y recopilar información acerca de los distintos sensores que suelen incorporar las plataformas robóticas cuadrúpedas en la actualidad para así evaluar si alguna tecnología derivada podría ser empleada por nuestro robot. Debido al enfoque dado, esta sección será esquemática con una breve descripción del sensor y su función en los robots, encontrando una revisión más exhaustiva en un artículo presentado en la 35ª *Chinese Control Conference*, en el año 2016 [40].

#### 3.4.1 Sensores de tiempo de vuelo

Este tipo de sensores funcionan enviando una onda electromagnética o sonora y calculando el tiempo que tarda esta en volver al sensor, en función de la velocidad de la onda, al ser reflejada en una superficie. De esta forma se obtiene una información de profundidad bidimensional del objeto en relación con la fuente. A continuación, se comentarán los más usados:

- **Sensores *ToF* por ultrasonidos:** son los más económicos y emplean una onda ultrasónica para localizar objetos. Los de uso con Arduino comunes (HC-SR04) tienen una distancia de medición de 2 a 40 cm con una precisión de  $\pm 5$  mm. Existen versiones industriales con hasta 10 m de alcance.
- **Sensores IR:** tienen un precio algo superior a los anteriores y se utilizan principalmente como sensores de distancia. Existen muchos tipos con distintos rangos, pero el más sencillo cuenta con un rango de detección de 2 a 30 cm en un cono de  $35^\circ$ .
- **Sensores *RADAR*:** estos sensores ven un uso muy limitado ya que se opta por los sensores tipo *LiDAR* que cuentan con mayor resolución y desarrollo. Existen desarrollos, pero generalmente son imitados con sensores de ultrasonidos junto a servomotores.
- **Sensores *LiDAR*:** presentan una amplia implementación y están disponibles en 2 rangos principales de precio en función de sus capacidades. Funcionan enviando un haz de luz y calculando el *ToF* hasta la vuelta con mayor resolución que los sensores IR y sin alteraciones del IR. Los más profesionales de iniciación, montados en proyectos similares, cuentan con motor para proporcionar una visión  $360^\circ$  a 8m y una resolución de 0.25mm. Su función principal es el mapeado de zonas.

### 3.4.2 Sensores ópticos (sin ToF)

Este tipo de sensores comprende aquellos que requieren una cámara para captar una imagen y procesarla. Quedan excluidos de esta categoría los que emplean haces de luz, pero no realizan una captura de una imagen. El funcionamiento general consiste en la extracción de información espacial a partir del procesamiento de una imagen con o sin información adicional. En este caso, existen dos dispositivos principales que se suelen equipar, ya que las cámaras térmicas son más especializadas:

- **Cámaras RGB:** existen varios enfoques posibles:
  - Uso de visión artificial para extraer información adicional de la escena tal y como hacemos los humanos.
  - Uso de visión estereoscópica para la localización espacial de objetos en imágenes sin profundidad.
  - Uso como sistema de teleoperación simple.

En cualquiera de los casos, existen sensores muy económicos con placas de control que incluyen *WiFi* y *Bluetooth*.

- **Cámaras de profundidad:** este tipo de dispositivos combinan la información de un patrón de haces infrarrojos con la lectura de un sensor convencional, para segmentar objetos en una escena. El precio es algo elevado, aunque presenta muy buenas características con alcances de 5 metros y al contar con sensores convencionales, proporcionan mayor funcionalidad. Suelen implementarse en lugar de los *LiDAR* en robots comerciales, cuando el mapeado no es un requisito.

### 3.4.3 Sensores de presión/fuerza y posición/velocidad

Este tipo de sensores se emplean para medir la variación o aparición de esfuerzos sobre ellos, en cuanto a la posición, pueden extraer el contacto y la posición de objetos cercanos o del mismo robot con respecto al mundo. Existen varios tipos de sensores en la robótica cuadrúpeda y especialmente en los sistemas no comerciales:

- **Sensores de fuerza/presión:** estos sensores constan de dos capas conductoras aisladas que, bajo presión, contactan generando una resistencia medible mediante una entrada analógica con *ADC*. Tienen un rango limitado 30-1000g y una precisión de  $\pm 3g$ . Existen alternativas más

caras como las galgas extensiométricas que aumentan el rango de medida.

- **Sensor de efecto Hall:** estos sensores son empleados principalmente para detectar superficies o posiciones ya que miden variaciones en el campo magnético estático. El precio es relativamente bajo, aunque son susceptibles al entorno.
- **Encoders o tacómetros:** existen de varios tipos y su propósito es conocer la posición exacta de un motor o accionamiento, así como poder estimar velocidades. Son más propios de la robótica convencional con un precio de adquisición alto.
- **Receptores GPSS:** aunque son mas comunes en los drones, hay implementaciones que cuentan con estos receptores de posicionamiento por satélite para ubicarse durante las labores de teleoperación.

#### 3.4.4 Sensores de orientación

Este tipo de sensores, también conocidos como *IMUs* (Unidad de Medición Inercial), se emplean para conocer la orientación en un espacio de coordenadas del robot con respecto al mundo. Esto lo hace al medir la velocidad, orientación y fuerzas gravitacionales empleando acelerómetros y giroscopios. Se pueden adquirir sin acelerómetro, pero suelen ir juntos al ser estos necesarios para corregir las desviaciones en las medidas de los giroscopios usando por ejemplo un control PID sobre la señal. Existen muchos tipos con más medidas o menos, pero su precio para una unidad estándar es bajo.

#### 3.4.5 Sensores de sistemas

En esta categoría entran todos aquellos sensores que son empleados para mantener un correcto funcionamiento del robot. Estos son:

- **Sensores de temperatura:** son empleados para medir la temperatura interna del robot y activar generalmente un sistema de refrigeración/ventilación. Su funcionamiento está ligado a la expansión y contracción de un material que afecta a la resistencia medida. También es común encontrarlos con sensores de humedad. Su precio es bajo.
- **Sensores de tensión:** formados por un divisor de tensión interno, estos sensores económicos, permiten medir la tensión que circula por un circuito. Su propósito suele ser la medición del nivel de la batería.
- **Sensores de corriente:** suelen contar con un sensor de efecto Hall que mide la variación en el campo magnético, producto de la inducción de la

corriente circulante. Su precio es muy bajo y suelen ser acompañados de relés para evitar que corrientes altas puedan dañar componentes en el circuito.

### 3.4.6 Motores y actuadores

Es fundamental para la locomoción, disponer de los motores y actuadores más apropiados para nuestra implementación. Aunque la mayoría de estos sistemas tiene puntos en común en cuanto a su método de funcionamiento, los resultados y el sistema de control necesario, así como el coste, son muy diferentes. A continuación, se recogen grosso modo algunos de los empleados, así como su uso en este tipo de proyectos:

- **Motores DC:** este tipo de motores hereda su nombre del tipo de corriente con la que operan, en este caso, directa. Su funcionamiento suele estar acotado a tareas que requieren un control más preciso del movimiento y existen 3 modelos principales:
  - **Servomotores:** tienen un gran uso en su variedad más compacta, generalmente para dirección en vehículos por radiocontrol. Permiten mucho par de fuerza en poco espacio y funcionan generalmente con un motor de corriente continua (CC) y una serie de engranajes a modo de reductor que generalmente se encuentran acotados para operar en unos grados determinados. Son los empleados para este tipo de plataformas robóticas. Su coste es medio, salvo en su versión industrial que cuenta con más potencia y coste.
  - **Sin escobillas o sin núcleo:** son los motores más comunes dentro de los de CC ya que al no contar con núcleo y, por tanto, tener menos inercia, permiten arranques y paradas más rápidas con menor peso. Suelen ir acompañados de reductores para obtener un par mayor.
  - **Paso a paso:** son los motores más precisos y generalmente cuentan con menos par de los servomotores. Son muy usados en sistemas de impresión 3D.

- **Motores AC:** este tipo de motores hereda su nombre del tipo de corriente con la que operan, en este caso, alterna. Su funcionamiento suele estar acotado a tareas que requieren un control menos preciso del movimiento más potencia o revoluciones y existen 3 modelos principales y suelen ir acompañados de reductores para obtener un par mayor:
  - **Universal:** este tipo de motores son capaces de recibir tanto CC como CA. Son de los más extendidos en pequeños aparatos como en electrodomésticos por contar con un par de arranque mayor a los de inducción junto con una elevada rotación. Suelen verse para simular un *LiDAR/RADAR*, en robots con ruedas y en robots con herramientas.
  - **Asíncrono:** este tipo de motor es el más común en la industria y tiene utilidad en propulsión de hélices y en algunos sistemas robóticos de investigación.
  - **Síncrono:** presentan una mayor precisión que los asíncronos y son empleados en aquellas situaciones en las que se requiere mayor control sobre posición.
- **Accionamientos piezoeléctricos:** este tipo de accionamientos son empleados para generar momentos o esfuerzos cuando el material es excitado por una corriente eléctrica. No requieren ningún tipo de mantenimiento y operan con corrientes bajas. Por contra, no ejercen mucha fuerza. Son especialmente interesantes en la micro o nano robótica y son vistos generalmente en investigación.
- **Accionamientos neumáticos:** este tipo de accionamientos utilizan válvulas o cámaras de aire que realizan un trabajo al aumentar la presión interna. Generalmente se utilizan en sistemas de investigación o robótica industrial ya que requieren de una bomba de aire para funcionar. No obstante, recientes investigaciones mencionadas en el capítulo 3.2, apuntan a su uso como sistemas homólogos a la musculatura humana, más eficiente y precisa que los motores industriales.

### 3.5. Sistemas de simulación y diseño

Este campo es quizás el más conocido, por lo que la investigación se centró en establecer qué podría aportar cada uno de estos programas o métodos al proyecto, y evaluar cuales eran los más usados en la actualidad para este tipo de proyectos.

#### 3.5.1 Gazebo [41]

Gazebo es un *software* de simulación 3D completo y gratuito para robots. Se emplea generalmente en *Linux* con paquetes de *ROS* (Sistema Operativo de Robots) tiene las siguientes características principales:

- Permite simular las dinámicas de movimiento con precisión empleando distintos motores de simulación.
- Permite incorporar distintos sensores y simular su iteración con el entorno.
- Inclusión de modelos de robots mediante *URDF* y *XACRO*.
- Permite la comunicación con la nube o por *TCP/IP* mediante sockets.
- Control vía línea de comandos para simulación y control de los modelos.

#### 3.5.2 RViz [42]

De forma similar a *Gazebo*, a menudo complementaria, *RViz* se emplea como simulador 2D de los robots y sus sensores. Principalmente es usado en tareas con navegación para simular mapas, así como para monitorizar la posición del robot a partir de los datos en tiempo real enviados por el mismo. Se suele emplear en conjunto con un paquete de *ROS*. Este *software* es gratuito.

#### 3.5.3 Coppelias Sim/V-Rep [43]

Se trata principalmente de un simulador 3D para evaluar estrategias de control en plataformas robóticas. Incorpora simulación de físicas y ce un uso principal en brazos robóticos ya que trabaja muy bien con ese tipo de cadenas cinemáticas y calcula de forma precisa simulaciones físicas de movimientos. Este *software* es gratuito para estudiantes.

#### 3.5.4 CAD/CAE/CAM [44]

Este tipo de programas permiten diseñar los componentes de un robot, así como evaluar los movimientos y esfuerzos en el modelo final. Cuentan con unas capacidades de simulación de control algo más limitadas que sus anteriores, pero ofrecen prestaciones superiores a la hora de simular esfuerzos y físicas. La gran ventaja de estos sistemas, como el *Autodesk Inventor*, es que todo el desarrollo se puede llevar a cabo con un único programa. Este *software* es gratuito para estudiantes de la Universidad de Alicante.

#### 3.5.5 ADAMS [45]

*ADAMS* es quizás uno de los *softwares* más potentes para realizar simulaciones físicas y dinámicas con robots, obteniendo una gran precisión en los valores simulados. Por el contrario, es un sistema extremadamente complejo y en cuanto a la simulación de arquitecturas complejas, requiere asistencia de otros programas como *Matlab*. Requiere una versión de pago con un coste elevado.

#### 3.5.6 Pybullet [46]

Se trata de un *software* de simulación 3D con físicas que a menudo se acompaña de entornos para aprendizaje automático con *Gym* de *OpenAI*. Es una plataforma que se encuentra aún en desarrollo, pero debido al auge del *machine learning* en la actualidad, está poco a poco ganando tracción. Está escrito en Python y está disponible para todas las plataformas de forma gratuita.

#### 3.5.7 Matlab + Simulink [47]

Se trata de la herramienta de cálculo más avanzada actualmente y cuenta con lenguajes de programación propios y multitud de paquetes que le permiten modelar y simular la gran mayoría de aplicaciones tecnológicas actuales. Su uso en robótica se da tanto para la implementación de algoritmos de control, como para su simulación, adquisición y procesamiento de datos una vez conectado al robot real por medio de cualquier sistema de envío de datos. Presenta gran potencial, aunque requiere un nivel intermedio de conocimientos para desarrollar

aplicaciones complejas. Este *software* es gratuito para estudiantes de la Universidad de Alicante.

## **3.6 Sistemas y métodos para la teleoperación y monitorización**

Este tipo de sistemas se emplean para dar instrucciones de movimiento a los robots, así como monitorizar el estado en el que se encuentran. Existen diversos y distintos tipos, especialmente en los sistemas profesionales (robótica espacial o submarina) por lo que, de nuevo, pondremos el foco en los empleados en la robótica de investigación, concretamente en prototipos similares al nuestro.

### **3.6.1 Teleoperación clásica con mandos de control**

Quizás este sea el método más común, ya que no requiere implementar ningún modelo o estrategia de control avanzada para el sistema robótico. Generalmente se relacionan las distintas entradas de estos dispositivos directamente con las salidas o comportamientos de los accionamientos de los robots. Existe además una implementación con una aplicación móvil que emula al mando [48].

### **3.6.2 Teleoperación avanzada con mandos de control**

Este tipo de implementación se encuentra menos establecida que la anterior en sistemas no comerciales, aunque con un principio de funcionamiento similar. En este caso se utiliza el mando para indicar movimientos y acciones y el sistema de control interno del robot planifica las trayectorias y movimientos en base a una estrategia de control.

### **3.6.3 Teleoperación o control autónomo**

Esta es la técnica es la menos desarrollada en este ámbito y consiste en, mediante algoritmos inteligentes y sensorización, indicarle al robot una posición a alcanzar y que este lleve a cabo todos los procesos necesarios para alcanzarla. Se suele ver en plataformas de investigación o en sistemas con navegación autónoma avanzada.



## 3.7 Sistemas y técnicas de impresión 3D

Debido a que el prototipo requiere que sus componentes estructurales sean impresos, ya que no se pueden adquirir de ninguna forma y serán modificados a medida, fue necesario realizar un estudio de los distintos sistemas, técnicas de impresión 3D y configuraciones, utilizados en la actualidad o recomendadas para este tipo de aplicaciones.

### 3.7.1 Técnicas de impresión 3D

Actualmente existen diferentes técnicas principales para la impresión de prototipos en 3D comerciales:

- Extrusión de Material (FDM)
- Polimerización VAT (SLA, DLP y SCL)
- Fusión de polvo (SLS)
- Inyección de Material (MJ)
- Inyección de Aglutinante (BJ)
- Deposición directa de energía (Direct Energy Deposition)
- Laminado en hojas (Sheet Lamination)

De los cuales se emplean principalmente las 3 primeras opciones debido a la sencillez, el coste y, al menos en el caso de las dos primeras, su accesibilidad económica.

#### 3.7.1.1 Extrusión de material o 'Fuse deposition modeling' (FDM)

Quizás el método más popular. El modelado de deposición fundida, o impresión 3D FDM, es un método de fabricación aditiva en el que las capas de materiales se fusionan en un patrón para crear un objeto. El material generalmente se derrite justo después de su temperatura de transición vítrea y luego se extruye en un patrón al lado o encima de las extrusiones anteriores, creando un objeto capa por capa.

Se pueden usar muchos tipos de materiales con las técnicas de *FDM*, incluidos los termoplásticos más comunes, chocolate, pastas e incluso materiales "exóticos" como termoplásticos con infusión de metal o madera. Normalmente se emplean máquinas cartesianas de tres ejes,

aunque existen distintas configuraciones y variaciones. Actualmente es ampliamente aceptado como la forma más sencilla de lograr la impresión 3D, ya que la *FDM* es barato y bastante eficiente dominando el mercado.

Una de las mayores ventajas de la impresión *FDM* 3D es la escalabilidad: se puede escalar fácilmente a cualquier tamaño siempre que quepa dentro del área de impresión, la cual también se puede escalar en ciertas impresoras modificándose. Otra ventaja es la flexibilidad del material. En cualquier impresora *FDM*, se puede imprimir una amplia variedad de materiales termoplásticos y filamentos exóticos con relativamente pocas actualizaciones y modificaciones, algo que no se puede decir de otros estilos en los que un material debe ser una resina o un polvo fino.

Sin embargo, una de las desventajas de la impresión 3D *FDM* a las que se hace referencia con mayor frecuencia, es la calidad o el detalle de las piezas. Debido a que el material debe extruirse en capas y tiene un cierto grosor predefinido por la boquilla, las impresiones de alto detalle son difíciles de lograr y, a menudo, requieren mucho procesamiento posterior para adquirir un aspecto acabado profesional. Otra desventaja de las capas en la impresión *FDM* es que crean un punto débil inherente en la impresión donde se une cada capa, lo que hace que las impresiones sean menos resistentes e inadecuadas para ciertas aplicaciones. Esto es algo que podremos, en la mayoría de los casos, paliar y corregir a la hora de preparar la pieza para la impresión.

#### 3.7.1.2 Polimerización VAT (SLA, DLP y SCL)

La impresión 3D de resina es, en lo referente a la historia de la impresión 3D, bastante antigua. La primera tecnología de impresión 3D basada en resina patentada se produjo en 1986 con la invención de la estereolitografía (SLA) de Charles "Chuck" Hall. Buscó curar las resinas poliméricas fotosensibles con luz ultravioleta, capa por capa.

Actualmente, para casi cualquier cosa, desde especificaciones de hilos hasta estándares de calidad, la Organización Internacional de Normalización (ISO) tiene un respetado conjunto de estándares de referencia que definen términos y principios. Esta establece que toda

impresión 3D basada en resina, en la que una resina de fotopolímero líquida se almacena en una tina (en lugar de inyectarse con tinta a través de una boquilla), se conoce colectivamente como polimerización en tina. Todos los demás nombres, como *SLA*, *DLP* o *LCD*, son subcategorías de esta tecnología.

Al igual que otras tecnologías de polimerización en cubas, ***SLA*** cura la resina fotosensible en capas para formar objetos 3D. La fuente de luz ultravioleta de *SLA* es un láser, controlado con precisión mediante espejos giratorios para "dibujar" cada capa que se imprime. Las principales ventajas de este método son una mayor precisión y una mejor calidad de impresión, ya que el movimiento preciso y el pequeño tamaño del láser permiten una resolución y un detalle más finos. Esta calidad mejorada presenta una desventaja principal, la velocidad de impresión, ya que extraer cada capa puede llevar más tiempo.

El procesamiento de luz digital (***DLP***) utiliza una fuente de luz ultravioleta diferente. En lugar de láseres, las impresoras *DLP* tienen proyectores UV, que funcionan mediante el uso de microespejos para controlar la luz proyectada. Las secciones transversales de cada capa se proyectan de una sola vez, curando todo en la capa a la vez. Con esto, pierde resolución en comparación con *SLA*, y los proyectores utilizados pueden ser voluminosos y costosos. Sin embargo, lo que sí obtiene es una velocidad de impresión mucho mejor, ya que se puede curar una capa completa a la vez.

Por último, tenemos la polimerización en cubas de ***LCD*** que es la más novedosa de las expuestas. En sus núcleos, todas las impresoras *LCD* 3D funcionan con el mismo principio: hacen brillar las secciones transversales de sus impresiones 3D utilizando una pantalla *LCD* para enmascarar una fuente de luz ultravioleta. Básicamente, esto reemplaza la configuración del proyector de *DLP* y la reemplaza con una pantalla más compacta y económica, a expensas de la resolución de impresión y la durabilidad de la impresora.

A continuación, se puede observar la comparación de los distintos métodos comentados en la Figura 21:

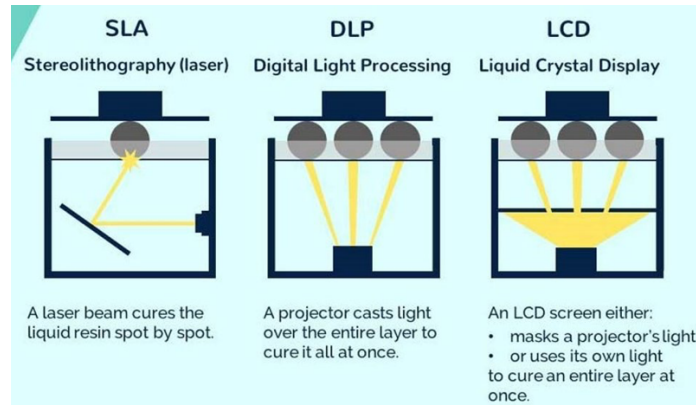


Figura 21. Comparativa de impresión con resina [49].

### 3.7.1.3 Fusión de polvo (SLS)

Las impresoras 3D de sinterización selectiva por láser (*SLS*) utilizan un láser y un polímero termoplástico en polvo o metálico para fabricar piezas. Debido al láser de alta potencia, generalmente se considera más complicado que *FDM* y *SLA*. En términos generales, el proceso de impresión comienza llenando el contenedor con una cierta cantidad de polvo de polímero. Luego, el contenedor se coloca en la máquina, donde comienza la etapa de calentamiento. Antes de imprimir, se utilizan calentadores para llevar el polvo a una temperatura justo por debajo de su punto de fusión. La impresión real comienza con el repintador depositando una sola capa de polvo sobre la plataforma de construcción.

Es ahí donde entra en juego el láser de CO<sub>2</sub>. La función del láser es inducir selectivamente la fusión entre partículas para formar un sólido en ubicaciones específicas. Los galvanómetros (similares a pequeños espejos) se utilizan para navegar el rayo láser a un punto específico en la plataforma de construcción. El láser se mueve a lo largo de un patrón de "punto a punto", solidificando toda el área de la sección transversal de una capa. Una vez que se completa la capa, el recubridor deposita una nueva capa de polvo y la plataforma de construcción desciende una capa en altura. Luego, el proceso se repite hasta que se completa la pieza.

Al igual que con cualquier otra tecnología de impresión 3D, *SLS* también tiene sus ventajas y sus desventajas. La mayor ventaja de *SLS* es que no hay necesidad de material de soporte adicional, independientemente de la geometría de la pieza. Esto es porque las

piezas se construyen dentro del depósito de polvo, el polvo no sinterizado actúa como material de soporte para la pieza impresa. Una vez impresa la pieza, se deja sin marcas del material de soporte en su superficie. Solo es necesario quitar el polvo con un cepillo. Además, debido a que las piezas se construyen con un láser que solidifica un área de sección transversal de cada pieza, se pueden imprimir fácilmente varias piezas a la vez. Esto hace que *SLS* sea ideal para pequeñas tiradas de fabricación. El objetivo de las impresoras *SLS* es llenar el área de construcción con tantas impresiones como sea posible para reducir el polvo de desecho no sinterizado. Además, en términos generales, el 50% del polvo no sinterizado se puede reciclar.

A diferencia de *FDM*, las piezas impresas con una impresora *SLS* no se pueden utilizar inmediatamente después de que se completa el proceso de impresión. Esto se debe a que las piezas deben enfriarse, lo que puede llevar mucho tiempo. La desventaja más significativa de *SLS* es que requiere tecnología muy novedosa y mucha potencia. Como tal, no está del todo lista para llegar al mercado de consumo, pero sí que hay máquinas comerciales.

### 3.7.2 Selección del método y material de impresión

Una vez conocidos los métodos y seleccionado el de *FDM*, es necesario centrarse en la selección del material óptimo para la impresión. Alternativamente podríamos tener acceso en la universidad a impresoras tipo, aunque el coste del material es más elevado.

En *FDM*, tal y como se ha comentado, existen distintos tipos de materiales que podremos usar en función de los resultados que esperamos de la Figura, desde deformables hasta resistentes a altas temperaturas. Para ayudar con la selección del material más apropiado, se utilizará el infograma de la Figura 22, que realiza una comparación general de los distintos materiales y su puntuación en una escala de 5 en función de su rendimiento por categoría.

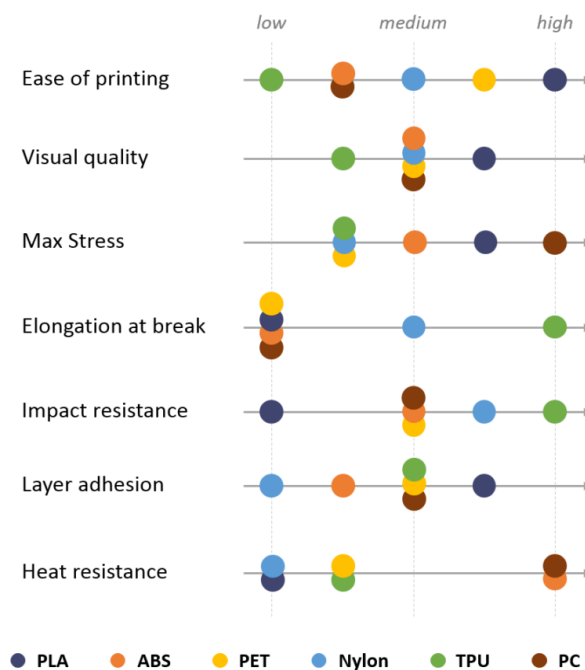


Figura 22. Comparativa de los materiales para FDM [50]

Tal y como podemos observar, para los requisitos actuales, los mejores candidatos son PLA, Nylon, ABS y PET. Sin embargo, debido a la dificultad de impresión del Nylon, este queda descartado. En cuanto al PLA, PET y ABS, podríamos sustituir perfectamente el PET por el PLA, pero debido a que el PET tiene tendencias a hilar en huecos pequeños si no se calibra bien la impresora, que su uso no está muy bien extendido y su gran sensibilidad a la humedad, el mejor candidato es el PLA, y el ABS como segunda. Además, son los más extendidos.

El ácido poliláctico o poliácido láctico (**PLA**) es un polímero o bioplástico constituido por elementos similares al ácido láctico, con propiedades semejantes a las del tereftalato de polietileno (PET) que se utiliza para hacer envases, pero que además puede ser biodegradable bajo ciertas condiciones a temperaturas del orden de 60°C. Se puede degradar en agua y óxido de carbono. Estos se producen principalmente de los desechos de maíz y, por tanto, en cuanto a su huella ecológica, es un buen candidato para un prototipado con muchas iteraciones de diseño. Además, es un producto seguro que no produce ningún tipo de irritación ni toxina al ser humano en cualquier forma de uso, con un coste medio de 20€/Kg.

El acrilonitrilo butadieno estireno o **ABS** es un plástico muy resistente al impacto muy utilizado en la automoción y otros usos tanto industriales como domésticos. Es un termoplástico amorfo. Se le llama también plástico de ingeniería o plástico de alta ingeniería debido a que es un plástico cuya elaboración y procesamiento es más complejo que los plásticos comunes, como son las poliolefinas (polietileno, polipropileno). Es un producto derivado del petróleo y que al ser fundido libera gases nocivos para el ser humano y, por tanto, requiere ventilación. Además de esto presenta problemas de impresión como el warping y requiere de temperaturas más elevadas lo que desgasta más las impresoras. Por último, se trata de un plástico no reciclable, aunque generalmente más barato que el PLA con un coste de unos 16€/Kg.

El **poliuretano termoplástico** (TPU) es un tipo de polímero elastomérico lineal termoplástico. Se trata de un elastómero caracterizado por su alta resistencia a las bajas temperaturas, abrasión y oxidación. Generalmente se utiliza para los recubrimientos de cableado y zonas en las que se necesita algo flexibilidad y resistencia. En los casos de uso con impresión FDM, se emplea para estructuras flexibles similares a la goma. Sus inconvenientes son: requiere una temperatura y calibrado de impresión muy superior para una correcta impresión, no puede ser extruido en las mismas formas que los plásticos anteriores, tiende a dejar hilos entre las piezas por lo que hay que imprimirlas de forma individual, requiere una temperatura más elevada y el coste es de unos 35€/kg.

### 3.7.3 Configuraciones básicas recomendadas

A continuación, se detallan los ajustes básicos para la impresión 3D tipo FDM con ABS y PLA:

- **Tª de extrusión:** PLA (190-220°C) 210°C como Tª recomendada y ABS (210-250°C) presentando 235°C buenos resultados; pero en ambos casos, se recomienda seguir las indicaciones del fabricante y ajustar hasta conseguir un resultado deseado pues según la pigmentación y calidad, las temperaturas óptimas son variables. Emplear una torre de temperatura para encontrar el valor óptimo puede ser una solución rápida y eficaz en caso de que las impresiones no sean satisfactorias.
- **Boquilla del extrusor:** se recomienda utilizar una boquilla de 0.4 mm de diámetro para poder imprimir a una altura de capa 0.1 mm en el caso de que necesitemos mucho detalle, o a 0.2 mm de altura que ofrece una calidad buena con mayor velocidad. Debido a los requisitos operativos de

las piezas, se desaconseja una boquilla mayor, aunque aumente la velocidad de impresión. Se recomienda, basándose en las tolerancias, una extrusión de 0.2 mm.

- **Temperatura de la cama caliente:** La temperatura de transición vítrea del ABS está en torno a los 105°C por lo que una  $T^a$  de 110-120°C es suficiente para ayudar con la adhesión, aunque se recomienda elevar la  $T^a$  unos 20°C más en las primeras capas para ayudar con el warping (es decir, cuando una capa se retrae sobre sí). También deberemos suministrar adherencia extra ya sea en forma de fijador en aerosol (laca), barra de pegamento o cintas adhesivas de doble cara (no deben de ser muy adherentes para no dañar o dejar restos en la pieza o en la cama). En el caso del PLA, su  $T^a$  de transición Vítrea está entre los 50-60°C, aunque, por norma general, podríamos imprimir con la cama en frío y fijación, se aconseja una  $T^a$  de 55°C. En cualquier caso, la necesidad de ayuda en la adherencia dependerá del material de la superficie de impresión.
- **Brim o raft:** las opciones de borde y balsa en la configuración de la adhesión a la cama proporcionan un contacto de superficie adicional entre la pieza y la cama. Esto ayuda con la adhesión de la primera capa y, al mismo tiempo, el borde o la balsa también actúa como una barrera sacrificable de material en caso de deformación (warping) en los bordes. De este modo, el borde o la balsa se verá afectada mientras la pieza requerida permanezca en buen estado. Para piezas muy pequeñas esto no es necesario, pero en nuestro caso, puede ser útil para mitigar los efectos del Elephant Foot con PLA y warping con ABS).
- **Altura de capas:** Generalmente se recomienda que la altura de la capa sea del doble del tamaño de la boquilla, en nuestro caso 0.8 mm. En lo referente a la altura de la primera capa, es aconsejable que esta sea un 150% del tamaño normal de la capa, es decir, 1.2 mm, para garantizar una buena superficie de adhesión.
- **Ventilación:** Se recomienda desactivar la refrigeración de capa hasta la número 10 para evitar el warping y mejorar la adherencia entre capas, así como una buena ventilación general sin corrientes de aire para que los gases desprendidos por el ABS puedan ser eliminados. En el caso de imprimir con PLA, se puede mantener durante toda la impresión. En ambos casos, esta será necesaria con piezas pequeñas o en capas rápidas donde el plástico no disponga de tiempo suficiente para solidificarse entre pasadas.



- **Capas laterales/bordes:** se recomienda emplear 3 capas en las paredes pues proporcionan la “coraza” de nuestras piezas y son las que soportan la mayor parte del estrés y son las responsables de la integridad estructural de las mismas. Podemos imprimir una pieza sin relleno, pero no sin paredes laterales. Aumentar estas hará que aumente la resistencia y el apoyo proporcionado por el relleno interno.
- **Post-procesamiento:** un baño de pulverización de acetona hará que la superficie pase de rugosa a lisa dejándola más uniforme y “soldando” pequeños fallos producto del error acumulado del movimiento. Esto solo recomendable con ABS, ya que el PLA no es igual de susceptible a la acetona.

En el caso del TPU el procedimiento es similar, pero aumentando la temperatura a 230°C, reduciendo la retracción a 1mm y la velocidad un 50% o no más de 20 mm/s. En cualquier caso, el uso de soportes es inviable y la configuración exacta requiere pruebas con el material escogido y la marca ya que existen distintas variaciones y calidades, cada cual, modificando los valores requeridos para una correcta impresión a diferencia de la poca variabilidad (generalmente por la pigmentación, del PLA y ABS).

#### 3.7.4 Patrón interno de relleno

En función del *Slicer* que tengamos, dispondremos de distintas configuraciones y patrones de relleno (Figura 23). Pero a nivel general podremos disponer de los siguientes patrones:

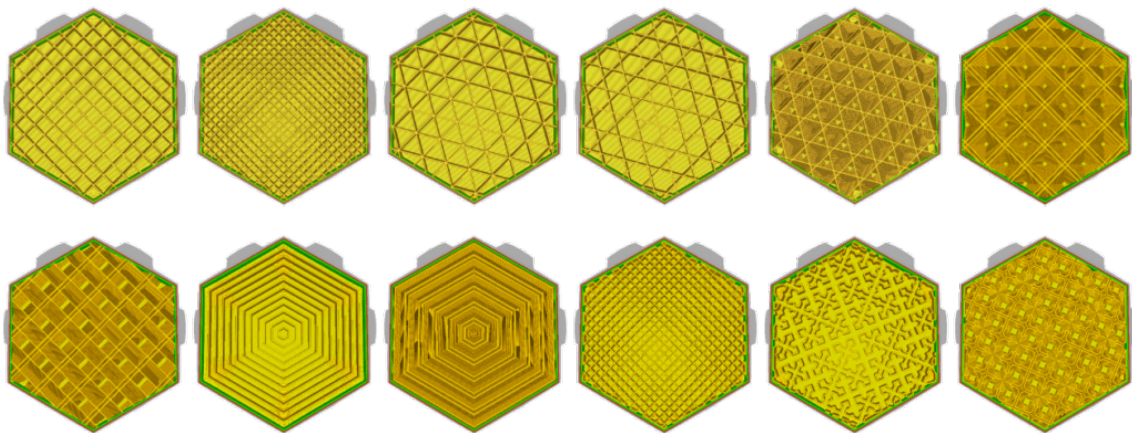


Figura 23. Comparativa de diversos patrones de relleno [51].

Estos patrones se pueden modificar con diversos parámetros hasta adquirir la distribución y grosor de paredes que mejor se ajusten a los resultados que queramos obtener. Por lo general, la configuración de estos parámetros se realiza de forma automática al configurar el perfil de impresión tal y como se ha indicado en la sección de configuraciones básicas recomendadas.

Sin embargo, uno de los parámetros que sí se deberá ajustar es el relleno interno. Este regula la distancia entre las líneas y el patrón de relleno (Figura 24), así como la densidad del relleno o cantidad de material interno. Este será fundamental para aumentar la resistencia, pero también lo hará el peso. Por tanto, se requiere un equilibrio. A continuación, se puede apreciar una comparativa de cómo cambia la densidad en función del patrón de relleno interno.

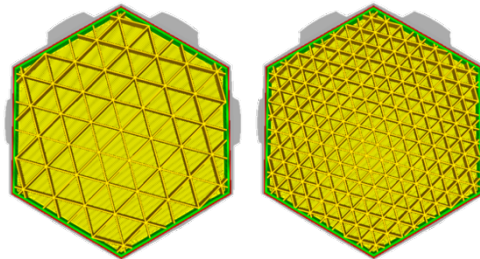


Figura 24. Comparativa de densidad de relleno interno [51].

Tal y como se puede apreciar, existe una gran variedad de patrones internos de relleno, así como de posibilidades de densidad interna. Con el objetivo de poder escoger el más idóneo para nuestro prototipo, se consulta una serie de gráficas (Figura 25) resultantes de pruebas de esfuerzo y carga en función de la geometría del relleno.

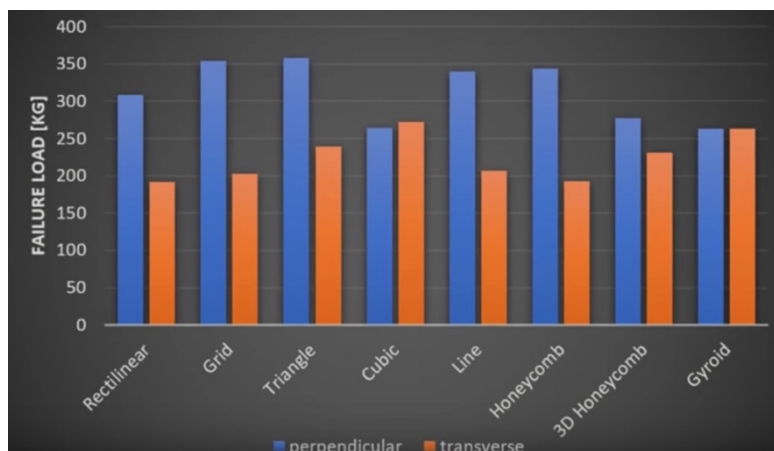


Figura 25. Comparativa de fallo a la carga según patrón interno y orientación [52].

A continuación, podemos ver otra prueba de esfuerzo (Figura26), esta vez sometiendo las piezas a compresión:

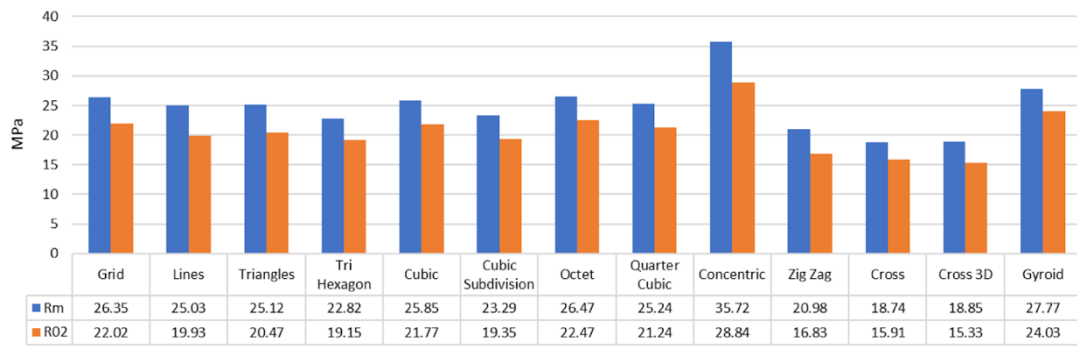


Figura 26. Comparativa de resistencia a presión según patrón interno [53].

Por último, se encuentra la comparativa de tiempo de impresión que también tendremos en cuenta (Figura 27), aunque en mucha menor medida y servirá para decidir en caso de empate:

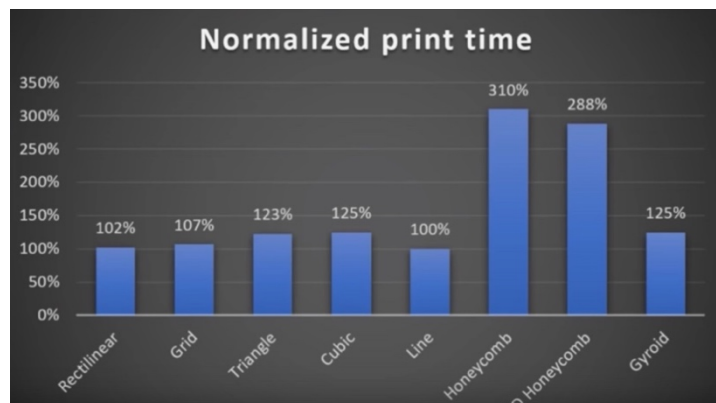


Figura 27. Comparativa del tiempo de impresión según patrón interno [52].

Tal y como se puede apreciar en ambos casos, los patrones cúbicos, concéntrico y giroide presentan los mejores resultados promedio ofreciendo resultados superiores al resto. En concreto, el patrón giroide, debido a su geometría singular, presenta la misma resistencia a carga independientemente del sentido de las capas y por ello es idóneo para nuestra aplicación en la que las fuerzas pueden venir en cualquiera de los ejes. Además, de los tres mejores, presenta el mejor tiempo de impresión.

Por otro lado, es necesario evaluar cómo afecta la densidad del relleno al comportamiento de la pieza (figuras 28 y 29), para ello, se puede utilizar la siguiente gráfica y tablas, productos de varios ensayos técnicos:



Figura 28. Comparativa general de la densidad del relleno frente al coste, calidad, velocidad y resistencia [54].

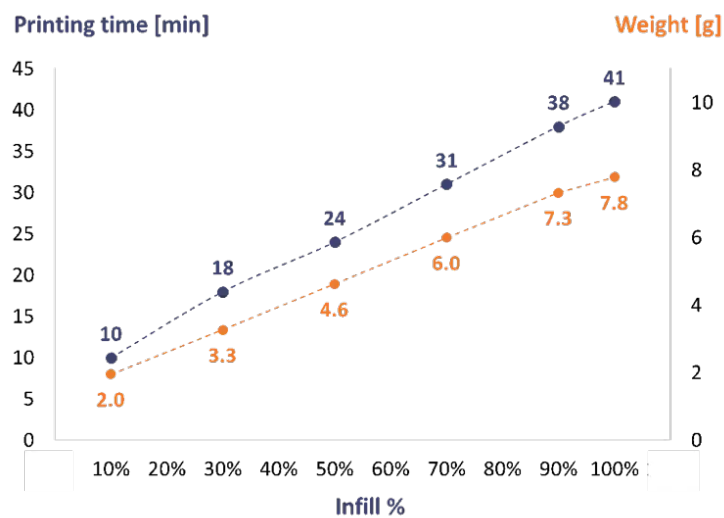


Figura 29. Comparativa de la densidad del relleno frente al peso y el tiempo de impresión [54].

---

## Capítulo 4: Metodología para el desarrollo

En este cuarto capítulo se pretende justificar las decisiones de implementación de los sistemas investigados en el capítulo anterior, como componentes del desarrollo práctico de este trabajo final de máster, así como los medios empleados para llevarlo a cabo. Para ello, se agruparán en los siguientes grupos las fases o hitos de desarrollo que son supeditadas entre sí.

### 4.1 Diseño e impresión del prototipo

Para elaborar el diseño se utilizarán dos programas: *FreeCAD* para exportar los diseños originales al formato *STEP* y de *Autodesk Inventor*, para crear y modificar los componentes necesarios. Una vez tengamos estos, se utilizará *Autodesk Inventor* para obtener un modelo 3D completo del prototipo sobre el que se podrá simular y extraer un fichero sobre el que basar nuestro modelo *URDF*. Se ha escogido *Autodesk* por su versatilidad y por la experiencia previa con el mismo y con *Fusión 360*. También se usarán de apoyo en modelos de los sensores y componentes encontrados en repositorios de modelos, así como de materiales de medición manual para tomar medidas sobre componentes reales. Por último, utilizaremos *Excel* y herramientas de cálculo para escoger los componentes electrónicos y calcular el par de los motores para usar los más adecuados.

Para la impresión 3D, se utilizará la herramienta de impresión o *Slicer Ultimaker Cura* por preferencia personal pues, en este caso, cualquier *Slicer* compatible con una impresora 3D sería válido. En cuanto a la técnica de impresión, se utilizará por una impresora *FDM*, concretamente la *Ender 3* con *PLA* y *TPU* en varios colores. Esto es así debido a que el post procesado por piezas es mucho menor, y el *PLA* es biodegradable, por lo tanto, si hay algún fallo de impresión, podremos reciclar la pieza, en cambio, con una impresora de resina o *ABS*, no podríamos hacerlo. Para calcular los costes de tiempo, plástico y económicos, se utilizará una tabla en *Excel* tras preparar los archivos.

Finalmente, el diseño del circuito electrónico se realizará mediante el programa de diseño de circuitos *Fritzing*. Este software permite diseñar componentes y exportar los diseños a distintos formatos, entre ellos, *PCBs* en el caso de querer compactar el sistema.

## 4.2 Construcción y pruebas básicas del prototipo

Para la construcción, se deberán seleccionar los materiales y herramientas necesarios para llevar a cabo el circuito, el ensamblaje y, posteriormente, las pruebas. Para realizar estas últimas, será necesario elaborar una serie de códigos en Arduino/C++ para comprobar el correcto funcionamiento de los sensores y componentes electrónicos. Se utilizará este programa por su facilidad de implementación y porque se podrá reutilizar parte del código empleado en el sistema final. Las pruebas las podremos realizar tanto con la placa ESP32, como con una Arduino *Uno/Mega/Nano* y la propia fuente interna proporcionada por la alimentación USB. En cuanto a la lista de sensores seleccionados, esta será justificada en el quinto capítulo, así como el sistema de actuación para el prototipo.

Para la elaboración de la guía de construcción y el proceso, se tomarán una serie de fotografías de los sucesos y problemas encontrados así de los pasos dados para obtener el prototipo, y se detallarán en un vídeo. Sí desarrollará una guía de construcción específica para aquellas partes modificadas o creadas, ya que su uso no está reflejado en ninguna documentación externa. En cuanto a los sensores y sistemas, se debatirán en su sección correspondiente del quinto capítulo, escogiéndose en base a lo visto en el capítulo tercero.

## 4.3 Control básico del prototipo

El control básico se hará en C y C++. El objetivo, por tanto, será de seguir los pasos vistos y emplear la ESP32 como controlador a bajo nivel y un sistema Linux como controlador de nivel alto que reciba las lecturas y señales de los sensores, así como controlar las articulaciones del sistema. Es decir, se pretende realizar un código base que permita posteriormente el control avanzado del robot. Aunque la placa es lo suficientemente potente, las funcionalidades brindadas por el apoyo de una *RPi (Raspberry Pi)* o un *PC* es muy superior. El método de comunicación será por *Wifi* y *Bluetooth* con la posibilidad de emplear un puente serie en el caso de que no se disponga de una conectividad inalámbrica y sea necesario incorporar un sistema *RPi* local, es decir, sobre el robot.

#### 4.4 Control avanzado, programación y simulación del prototipo

Tras la implementación inicial del control básico, se tratará de realizar un control más avanzado del robot en simulación, implementando el método de marcha con las curvas de Bézier en lazo abierto visto en el punto 3.3.4, pues ofrece los mejores resultados de los disponibles. Se tratará de implementar además su algoritmo de aprendizaje por refuerzo para nuestra versión de Spot Micro a modo de generador de trayectorias. Por lo tanto, se necesitará conectar con un sistema *ROS* (a bordo o remoto) para procesar lo captado por el controlador y realizar los cálculos de alto nivel para la generación de las trayectorias de cada articulación para una futura implementación en el robot real. Esto permitirá también monitorizar el sistema y evaluar los rendimientos de cada componente en tiempo real.

En cuanto a la programación, se hará en C, C++ y Python enfocada a *ROS* para la gestión de la información. Por último, para la simulación emplearemos 2 plataformas: Inventor para construir el modelo completo y Pybullet, para realizar el entrenamiento por refuerzo del modelo de control del prototipo. Se han escogido ya que son las únicas que permiten estas funcionalidades de las arriba descritas y, en el caso de Rviz, no se descarta una posible incorporación como apoyo o versión simple de Gazebo.

El objetivo por tanto es obtener una plataforma para la simulación de nuestro robot que permita en un futuro ser implementada sobre el robot real con otros algoritmos de navegación. En el caso de que fuese posible, como objetivo adicional, se tratará de comenzar a integrar lo obtenido en la simulación en el robot real.

## Capítulo 5: Implementación y desarrollo del prototipo

En este quinto capítulo se describirá el proceso de implementación y desarrollo de los distintos componentes del prototipo propuesto (Figura 32). Se implementará un sistema con un marco de control jerárquico para la locomoción y con una teleoperación asistida por un modelo entrenado con RL en simulación. Este tipo de control teleoperado consta de 2 partes principales: sistema maestro (online/local), y el sistema esclavo (local). A continuación, se puede observar un esquema general de este sistema de control (Figura 30), así como el algoritmo *GMBC* (Figura 31) seleccionado para llevar a cabo el control y el diagrama de la implementación en el prototipo. Los algoritmos para el aprendizaje automático y el control de cada extremidad de forma individual se encontrarán en el capítulo 5.7.7.

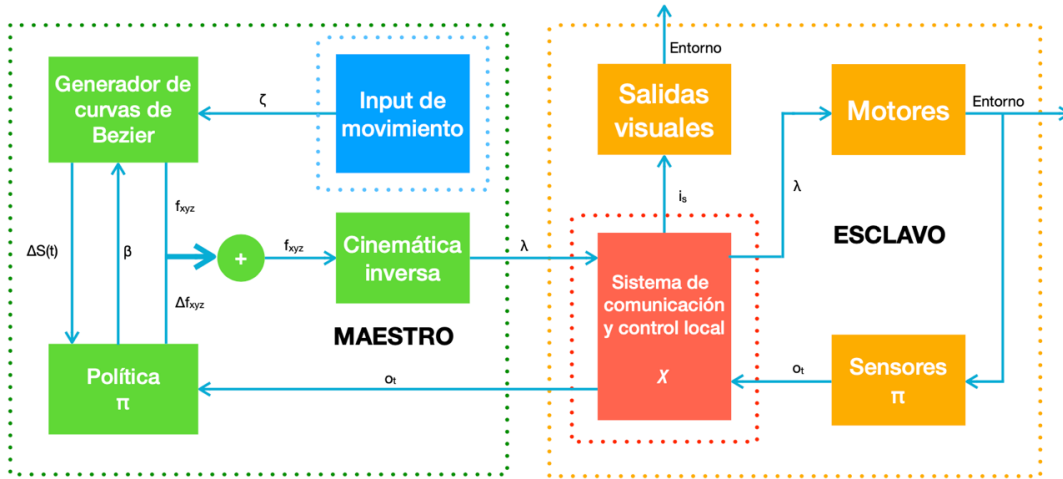


Figura 30. Esquema de control del sistema propuesto (Fuente propia).

---

### Algoritmo GMBC: Modulación de la marcha con curvas de Bezier

---

**Dada:** una política  $\pi$  con un parámetro  $\theta$ , el generador de curvas de Bezier  $\Gamma$ , un comando de movimiento externo  $\zeta$ , las observaciones de los sensores del robot  $o_t$  y la fase de la pierna  $S(t)$ .

1. Obtener la modulación del paso a partir de  $\pi$  con el parámetro aprendido  $\theta$ 
    - $\Delta f_{xyz}, \beta = \pi(o_t, \theta)$
  2. Calcular el posicionamiento para el paso en la curva de Bezier
    - $f_{xyz} = \Gamma(S(t), \zeta, \beta)$
  3. Devolver  $\Delta f_{xyz} + f_{xyz}$  al robot para el cálculo e la cinemática inversa
- 

Figura 31. Algoritmo GMBC [34].



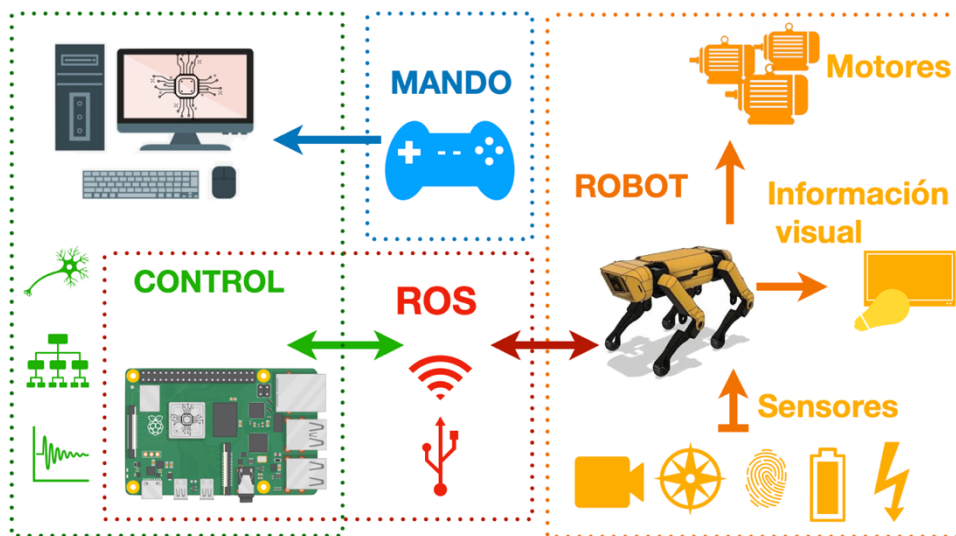


Figura 32. Esquema del sistema propuesto (Fuente propia).

A continuación, se dividirá la implementación en las distintas partes en función del sistema al que pertenecen y se explicará la unión entre las mismas, resultando en el prototipo final del sistema propuesto en este trabajo. Estas subdivisiones o bloques principales de desarrollo son, de acuerdo con las etapas o fases de desarrollo, *hardware*, verificación *HW* y *software*; es decir, la misma distribución realizada en la metodología del capítulo cuarto. No obstante, tal y como se explicó en la metodología, el objetivo inicial era una división en tres fases de desarrollo. El objetivo de esta división era conocer a fondo el prototipo y obtener la mejor versión posible del mismo dadas las limitaciones al concluir con el desarrollo. Por tanto, estas divisiones comentadas en el capítulo cuarto estarán presentes en la medida en la que se hayan desarrollado, en las fases siguientes:

- **Fase 1:** implementación de un prototipo en la misma línea que los robots *Spot Micro* desarrollados. Es decir, aprender y conocer a fondo la base sobre la que están trabajando todos los proyectos actuales, con componentes seleccionados a medida en base a las versiones estudiadas.
- **Fase 2:** implementación de una serie de mejoras en los componentes internos, principalmente de locomoción, así como propuestas o cambios de diseño para incorporar nuevas funcionalidades. El objetivo es mejorar y adecuar aún más el hardware para su uso final.
- **Fase 3:** implementación de sensorización completa y rediseño de los componentes necesarios para la versión definitiva. Esta tiene como objetivo enlazar con el sistema maestro y ser la mejor plataforma de

desarrollo para Spot Micro, manteniendo un coste de desarrollo similar al original, pero expandiendo enormemente sus funcionalidades.

Finalmente es necesario remarcar el hecho de que el desarrollo de las fases en este capítulo se basa en una progresión acumulativa, es decir, si no hay una variación o iteración sobre el desarrollo realizado en una fase anterior, este no será mencionado en las sucesivas fases y, por tanto, se aplica a la última versión. De esta forma, se sigue el orden de desarrollo del prototipo de forma cronológica, aportando justificación y orden a las decisiones tomadas entre las distintas fases.

## **5.1 Diseño e impresión del prototipo: Fase 1**

### **5.1.1 Obtención de modelos editables para Autodesk**

En primer lugar, se descarga el repositorio [8] en su versión del 18 de octubre de 2020. Esto es importante ya que se ha colaborado en el repositorio e iterado sobre esas versiones iniciales, en base a las pruebas y *feedback* obtenidos en el transcurso de las fases 1 y 2. Por lo que la versión actual del repositorio no es reflejo de la inicial sino un resultado de trabajo conjunto progresivo. También será necesario descargar e instalar el software *FreeCad* [55] para poder procesar los archivos “FCStd” y extraer los archivos “STEP” compatibles con *Autodesk Inventor* y *Fusión360*. Aunque estos últimos, no contengan las trazas y medidas originales, sí conservan las geometrías y extrusiones como componentes de un ensamblado y permiten trabajar sobre ellos a diferencia de los “stl” (ideales para imprimir) que son importados como una malla triangular no editables, visible en la Figura 33. El procedimiento es puramente mecánico: se deberá abrir cada archivo por separado y exportarlo al formato correcto tal y como se puede ver en la Figura 34. Una vez obtenidos los ficheros, podremos proceder al siguiente apartado.

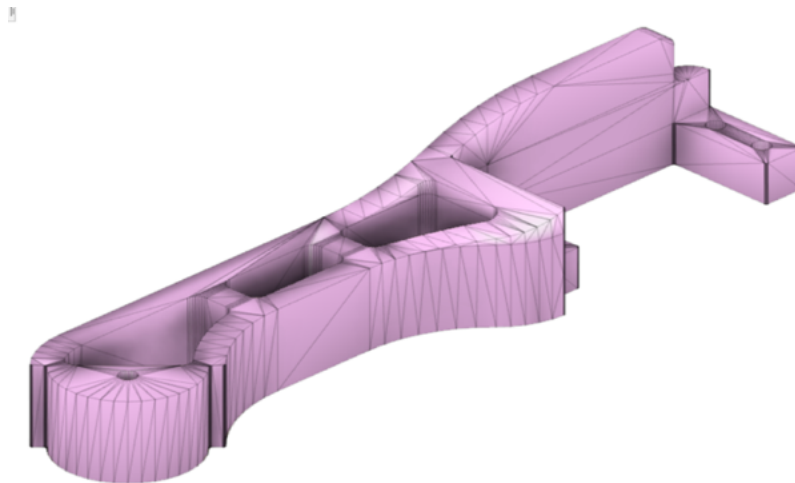


Figura 33. Importación del archivo “wrist.stl” original en Autodesk (Fuente propia).

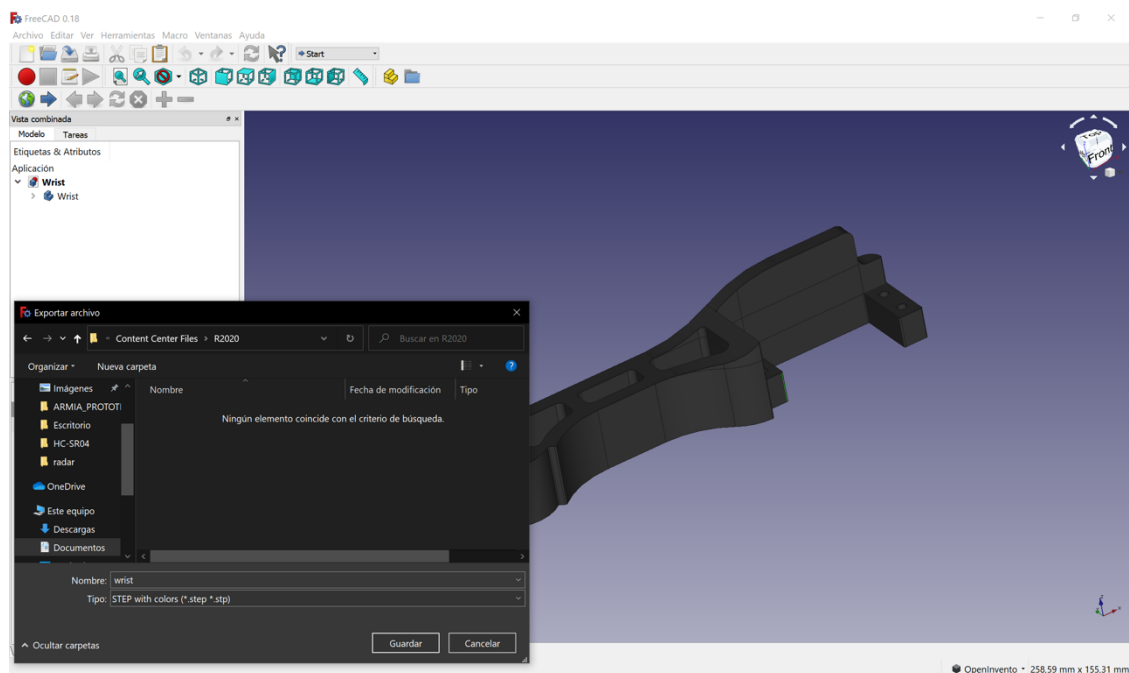
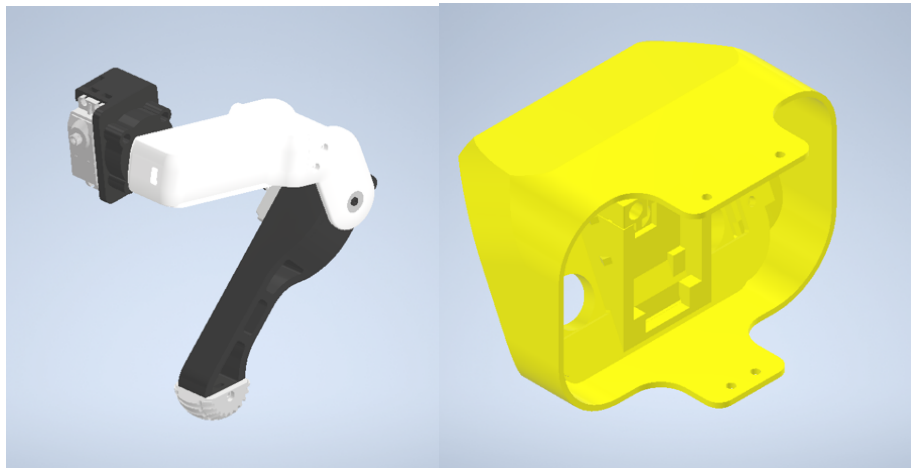


Figura 34. Exportación del archivo “wrist.FCStd” original al formato “STEP” (Fuente propia).

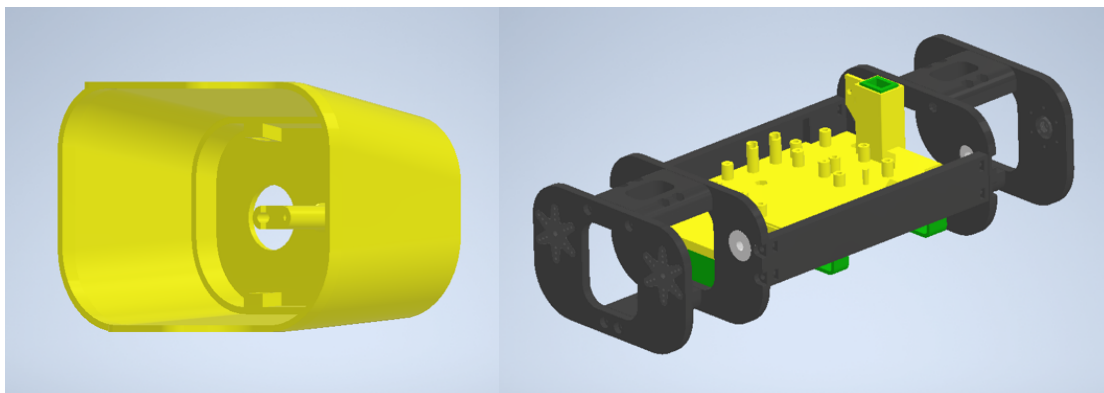
### 5.1.2 Ensamblado y comprobación del modelo virtual

Obtenidos los modelos en formato STEP, el siguiente paso es generar un ensamblaje en *Autodesk Inventor* para comprobar que todas las piezas encajan y estudiar los límites de los movimientos. El procedimiento óptimo para llevar a cabo esto es mediante una subdivisión de componentes, es decir, se harán subensamblajes de: dos de las patas, el cuerpo principal, el frontal y la parte trasera del robot.

El proceso para realizarlos puede encontrarse de forma detallada en multitud de sitios web, pero, en resumidas cuentas, consiste en la importación de archivos “IPT” (en nuestro caso generados automáticamente al importar el “STEPfile”), sobre un archivo de ensamblaje (IAM). Una vez importados en este, haciendo uso de las herramientas de restricciones de superficie, fijaremos entre si los distintos componentes, dejando libres de cualquier tipo de restricciones, los grados de libertad propios del robot. De este modo, se obtiene un modelo ensamblado de la misma forma que si incorporase las medidas de fijación reales, tornillería y tuercas también incorporables, pero para su simulación. A continuación, se pueden observar los ensamblajes completados de estos subcomponentes (figuras 35 a 38) en sus diferentes archivos “IAM” y la versión final obtenida junto a su homóloga de LibreCAD (figuras 39 y 40). Por último, se incorporará un modelo de servomotor RC obtenido de *GrabCAD* [56] para completar el ensamblado:



Figuras 35 y 36. Pie ensamblado (izquierda) y frontal ensamblado (derecha) (Fuente propia).



Figuras 37 y 38. Parte trasera ensamblada (V2) (izquierda) y cuerpo ensamblado (derecha) (Fuente propia).

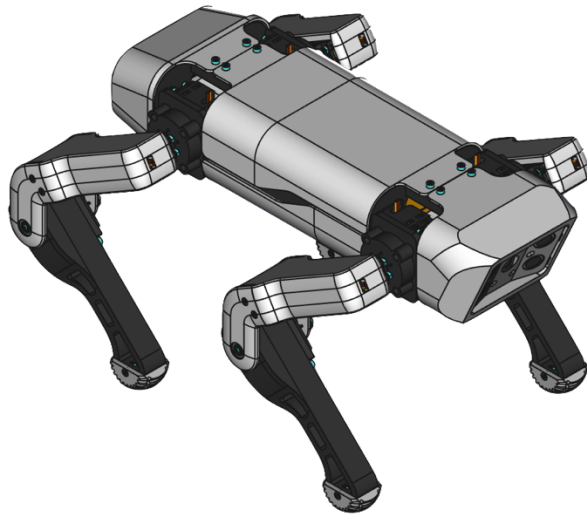


Figura 39. Robot ensamblado original [8].

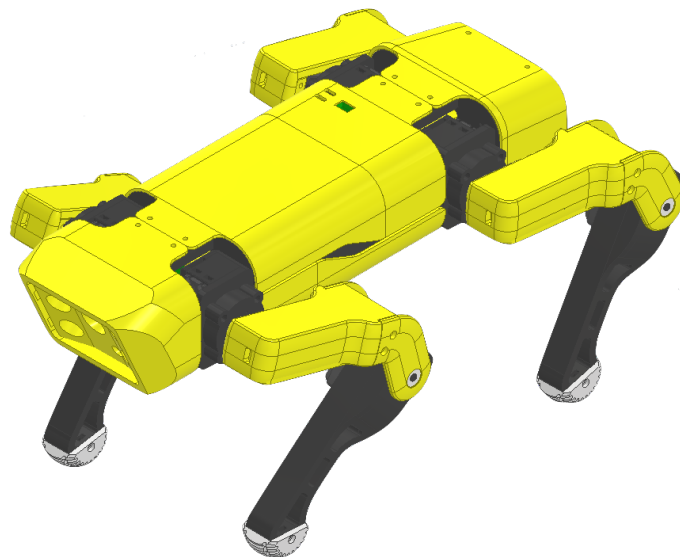


Figura 40. Ensamblaje en inventor de la fase 0 (Fuente propia).

En lo referente a la movilidad, se puede verificar que, tras establecer los límites correspondientes a los grados de libertad, el modelo funciona correctamente. A continuación, se pueden apreciar en la siguiente imagen (Figura 41), un compuesto de diferentes posiciones. Tanto el vídeo de las pruebas de movimiento del modelo como los archivos se encuentran disponibles en la [carpeta de google drive](#).

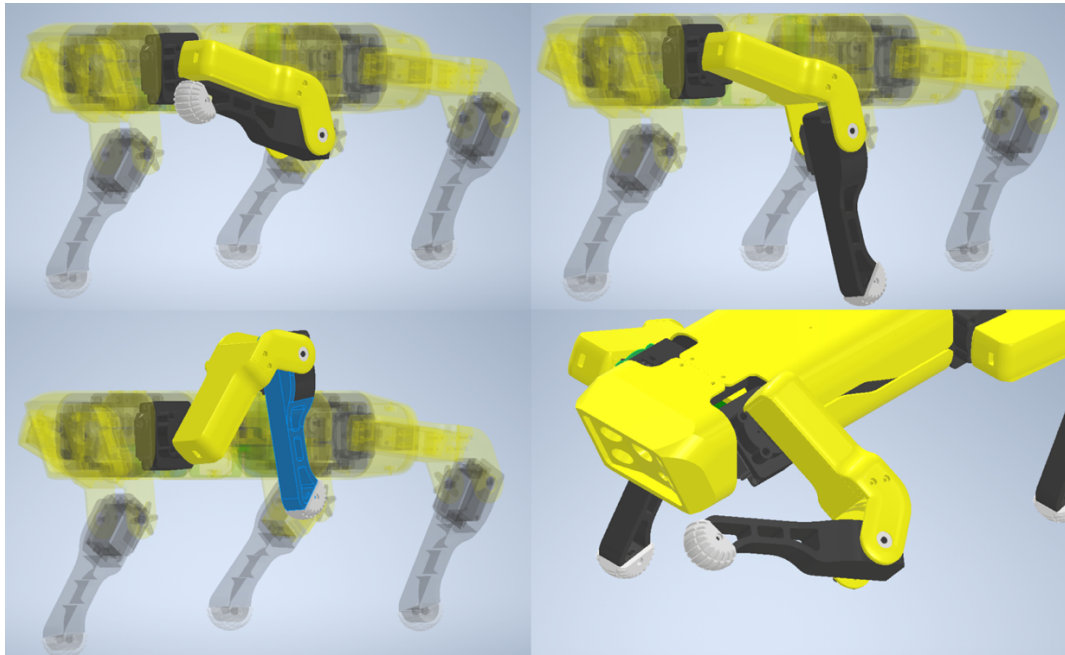


Figura 41. Pruebas de movimiento del modelo de la fase 1 (Fuente propia).

### 5.1.3 Estudio inicial de la cinemática del robot

Para el estudio de la cinemática, se utilizará el modelo disponible en *FreeCad* o *Autodesk Inventor* y sobre el mismo, se realizarán una serie de cotas para obtener medidas. De estas cotas, se extraerán los valores correspondientes a la cinemática para cada una de las articulaciones de las piernas que se rigen por simetría. Gracias a que la batería supone aproximadamente el 12% del peso, y los componentes electrónicos se hayan mayoritariamente en el centro y uniformemente en las articulaciones, se puede suponer que el centro de la placa como centro de masas del robot. Las imágenes de las que se puede extraer esta información son las siguientes:

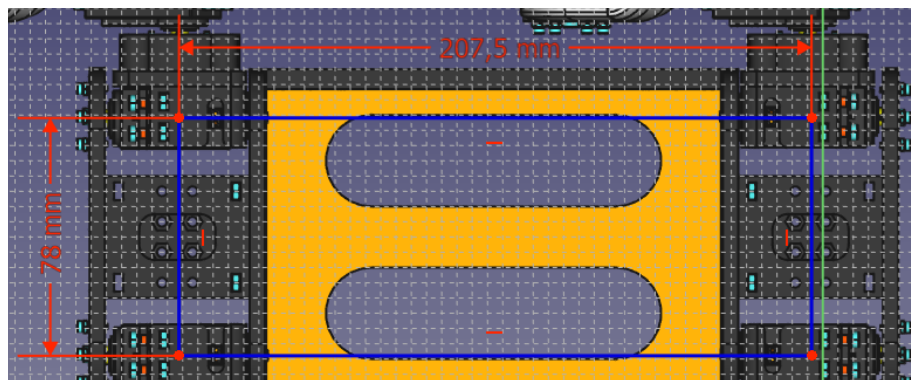


Figura 42. Medidas del cuerpo del robot (Fuente propia).

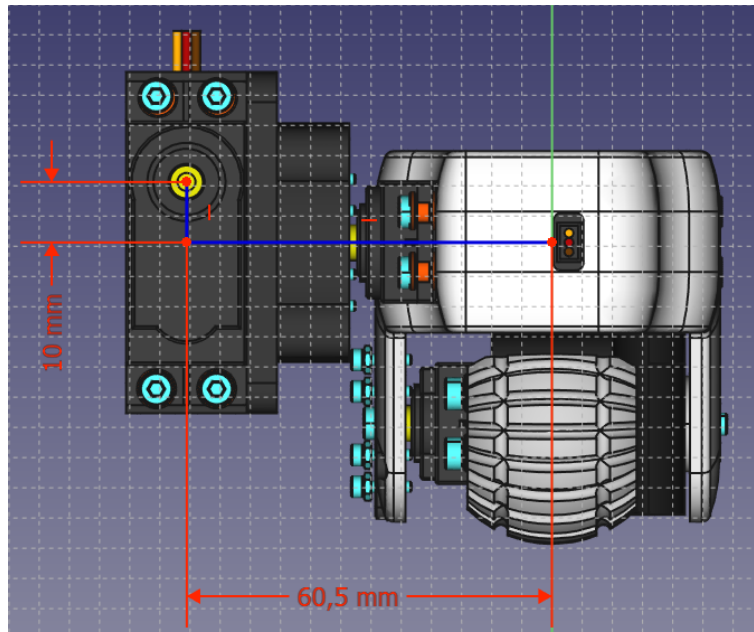


Figura 43. Medidas del hombro del robot (Fuente propia).

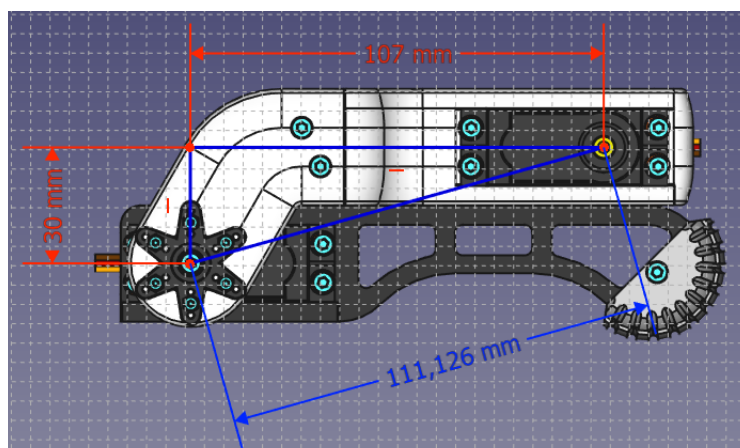


Figura 44. Medidas del miembro superior del robot (Fuente propia).

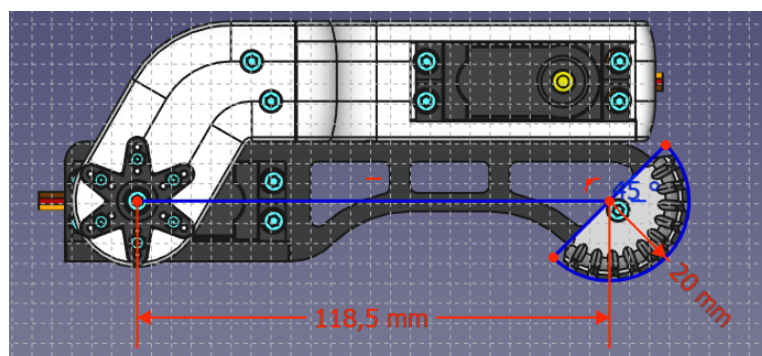


Figura 45. Medidas del miembro inferior del robot (Fuente propia).



Tal y como pudimos extraer del estado del arte, conocer o aplicar la cinemática inversa para este tipo de robots no es eficiente debido a la complejidad del controlador. Por tanto, es necesario desarrollar la cinemática inversa para el robot. Para ello se utilizarán los diagramas de las cadenas cinemáticas simplificadas de las figuras 46 y 47, se puede desarrollar un código que permita calcular la cinemática inversa para cada una de las patas del robot, dada una posición a alcanzar. Este método es más sencillo ya que el input necesario es una posición y el output, son 3 ángulos correspondientes a la posición requerida para cada uno de los servos de esa cadena cinemática, para lograr una determinada posición. El código en cuestión está disponible en el primer anexo en su versión definitiva tras su revisión en el apartado 5.7.6, en el que será explicado. Esto es así ya que no es necesario para la fase 1 de desarrollo, aunque sí conocer la cinemática para comandar posturas al robot.

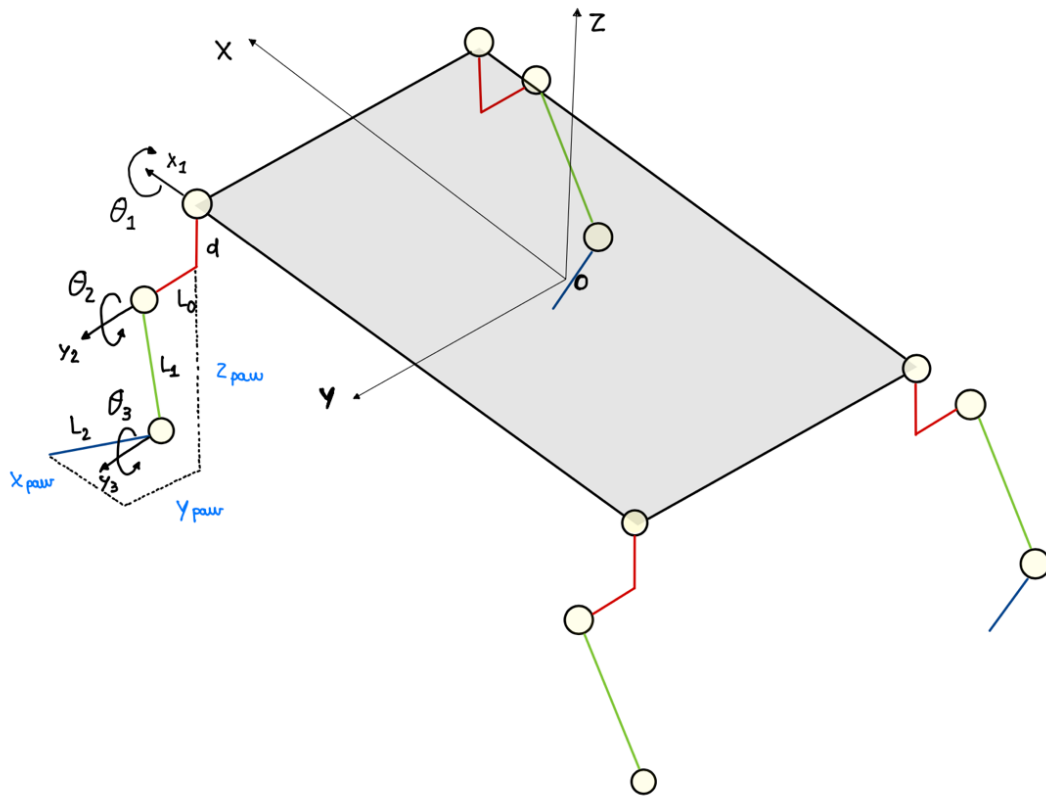
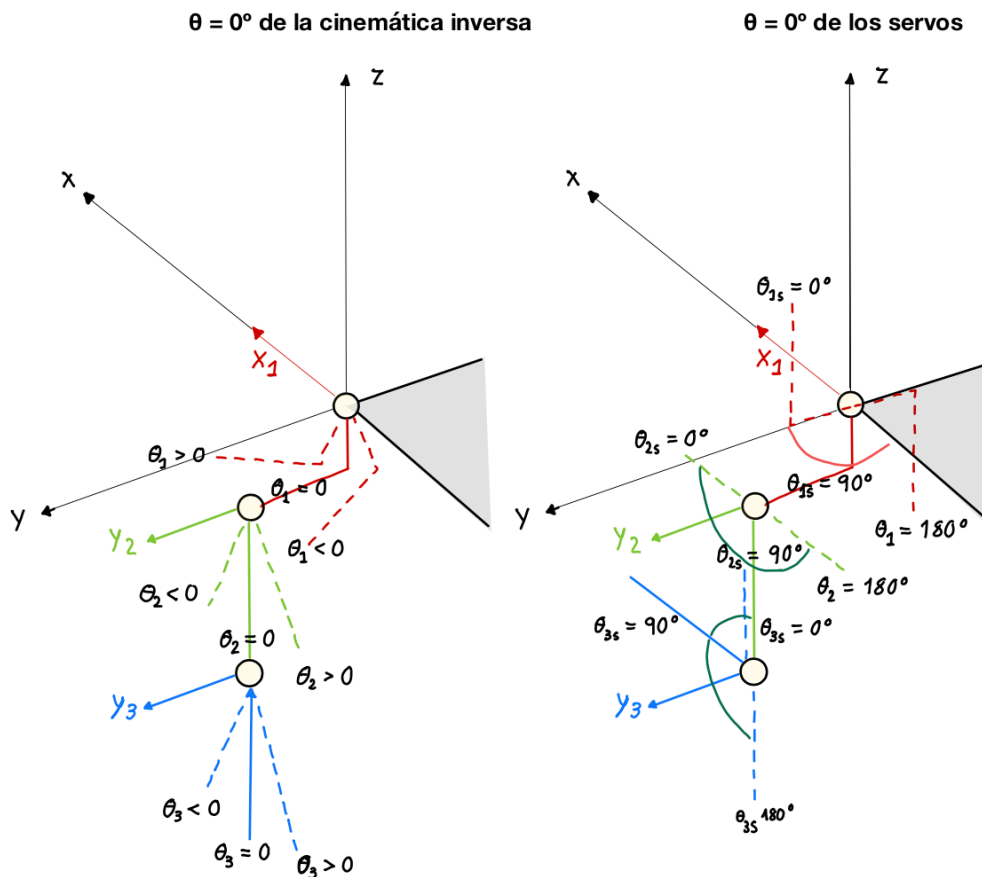


Figura 46. Cadenas cinemáticas de las patas de Spot Micro (Fuente propia).





Figuras 47 y 48. Cadenas cinemáticas de las patas de Spot Micro (Fuente propia).

#### 5.1.4 Elección de los componentes electrónicos y sensores

Una vez verificado el funcionamiento teórico del modelo, es necesario seleccionar los componentes electrónicos y sensores. En un principio, el equipamiento propuesto y compartido por la gran mayoría de implementaciones, es funcional y permite su desarrollo. No obstante, se evaluarán estos componentes uno a uno a continuación para verificar su usabilidad en la implementación.

En cuanto a los **sensores iniciales**, se seleccionaron 4:

- Dos **sensores de ultrasonidos** HC-SR04: estos van en la cabeza del robot y servirán para evitar obstáculos a 40cm
- Un **sensor inercial** MPU-6050 de 6 GdL: este sensor dará la orientación del centro de masas del robot y ayudará en el equilibrio.

- Una **cámara ESP32 Wifi**: esta irá en el morro del robot y permitirá visualizar su campo visual durante la teleoperación. También es capaz de usarse para visión artificial y dispone de algunos pines libres para hacerse cargo de algún sensor.

En lo referente a los **componentes electrónicos**, inicialmente son necesarios los siguientes para la fase 1:

- **Placa de control ESP32 de 38 pines**: será la placa con la que se controlará el robot y con la que finalmente se implementará el control a bajo nivel. Servirá para probar el correcto funcionamiento de los sensores.
- **Relé de potencia simple**: este componente es necesario para cortar la corriente de los motores si se detecta que algo no va bien.
- **Dos reductores de potencia DC-DC**:
  - **XL4016 DC-DC 9 A**: necesario para reducir la tensión de entrada de la batería y alimentar a los servomotores a 6,5V. Es razonablemente grande y permite un paso de corriente de 9ª.
  - **LM2596**: este es un reductor más pequeño con el que se bajará la potencia a 5V necesarios para alimentar la mayor parte de la electrónica.
  - Reductor a 3,3V interno de la ESP32: incorporado en la propia placa, servirá para la alimentación de los pocos componentes que utilicen 3,3V.
- **Placa de control de servomotores PCA9685 Placa PWM de 16 canales y 12 bits**: con esta placa se podrá controlar todos los servomotores desde un único canal I2C con una resolución de 12 Bits. Soporta hasta 6V de alimentación propia y con protección para los servomotores. Es capaz de soportar realmente hasta unos 9V tras ver casos en la red, por lo que los 6,5V de los servomotores clásicos de RC no serán un problema:
- **Dispositivo de interfaz serie adaptador FT232RL USB-TTL**: este será necesario para programar la cámara pues no tiene puerto USB propio.
- **Botón con enclavamiento y led de 19 mm y 30ª**: este es necesario para encender el robot y conocer su estado.
- **Conectores hembra mini USB**: estos serán unidos a dos cables internos conectados a su vez a ambas ESP32 para poder programarlas sin necesidad de abrir al robot.
- **Pantalla TFT de 32 bits**: se podrá mostrar información acerca de la cámara o del estado del robot.

Para seleccionar la **fente de alimentaci3n**, generalmente se estima en funci3n de la duraci3n de la operaci3n y la tensi3n e intensidad requeridas por los componentes que integran la plataforma rob3tica. No obstante, para este caso, el dise1o/tama1o solo permite baterías de 6200 mAh o menos. El objetivo es que la duraci3n de la operatividad sea m1xima por lo que una batería con m1s de 6.5V (tensi3n est1ndar m1xima que puede alimentar la placa de control de los servomotores) y una capacidad de descarga de A/h de al menos 30C es suficiente. La batería en cuesti3n es una LIPO 2S de 6200 mAh (ya que el gramaje con 5200 mAh era similar) con 80 C y 7.4-8.4 V de tensi3n de salida. El peso de la batería y el m1ximo aprovechamiento del espacio disponible hace que sea beneficiosa tanto para regular el centro de masas del robot, como para proporcionar la mejor alimentaci3n con margen para la ampliaci3n del equipo.

En cuanto a los **motores**, bas1ndose en lo aprendido en el Estado del arte, se tendría como mejor opci3n un motor sin escobillas de corriente continua con un reductor planetario o harm3nico. Sin embargo, debido a las limitaciones de tama1o y potencia, la mejor opci3n es la ya implementada por la mayoría de los usuarios, servomotores compactos de tipo “hobby”. Estos modelos ofrecen un buen par de potencia en comparaci3n a su tama1o y peso, al mismo tiempo que mantienen unos consumos de potencia bajos. Como extra a1adido, no requieren ninguna modificaci3n a los archivos del robot. Bas1ndose en la informaci3n proporcionada para la estimaci3n del peso (pl1stico) se obtiene un peso medio de 955 g aproximadamente (ver punto 5.1.7). Sin embargo, al tener en cuenta la suma de los componentes arriba descritos, el peso estimado para el cuerpo principal es de 1.9 kg.

Para evaluar qu3 servomotor es apropiado para la aplicaci3n, se puede hacer un sistema de ecuaciones conociendo las f3rmulas [57] para el momento ejercido por el motor ( $\text{fuerza} \cdot \text{distancia}$ ), tambi3n llamado torque. La ecuaci3n es muy sencilla y para realizar los c1culos, se necesitar1 saber los pesos de cada una pata, el del cuerpo, un valor del peso del servomotor que llamaremos “p” y las distancias entre ejes obtenidas en el estudio de la cinem1tica. Hay 3 servomotores por pata, y por tanto el orden jer1rquico de carga a soportar o de momento necesario, aumentar1 cuanto m1s cercano se encuentre el servomotor al punto de apoyo. Los nombres para estos servomotores ser1n: SHA (servomotor del hombro abductor), SHFE (servomotor del hombro flexor/Extensor) y SC (servomotor del codo). Los nombres difieren de la literatura que hace referencia a estos como las posiciones de hombro, codo y mu1eca.

Anatómicamente estos no serían los nombres apropiados pues, los grados de libertad 2º y 3º de cada pata, vistos en la cinemática del punto anterior, conforman el sistema de dos grados de libertad del hombro, y el 1º, el homólogo del codo. Además, el posicionamiento en el robot respalda este nombramiento.

En primer lugar, se calcularán los pares estáticos para cada pata suponiendo un apoyo completo de las 4:

- $X_n$  = peso del sistema hasta el punto en la cadena cinemática donde se acopla el servomotor.
- $y$  = peso del cuerpo del robot = 1.9 kg.

$$M_{SHA} = (y)/4 \cdot \text{sqrt}((20.75/2)^2 + (7.8/2)^2) = 5.26 \text{ kg}\cdot\text{cm}$$

$$M_{SHFE} = (y/4 + x_1 + p) \cdot 6.05 = (y/4 + 0.054 + p) \cdot 3.525 = 1.95 + 3.525p \text{ kg}\cdot\text{cm}$$

$$M_{SC} = (y/4 + x_2 + p) \cdot 11.12 = (y/4 + 0.120 + 0.054 + 2p) \cdot 11.12 = 7.49 + 11.12 \cdot 2p \text{ kg}\cdot\text{cm}$$

Conocidos estos datos, se hará lo mismo para calcular el par en estático para 2 puntos de apoyo:

$$M_{SHA} = (y)/2 \cdot \text{sqrt}((20.75/2)^2 + (7.8/2)^2) = 11.08 \text{ kg}\cdot\text{cm}$$

$$M_{SHFE} = (y/2 + x_1 + p) \cdot 6.05 = (y/2 + 0.054 + p) \cdot 3.525 = 3.72 + 3.72 \cdot p \text{ kg}\cdot\text{cm}$$

$$M_{SC} = (y/2 + x_2 + p) \cdot 11.12 = (y/2 + 0.120 + 0.054 + 2p) \cdot 11.12 = 13.05 + 11.12 \cdot 2p \text{ kg}\cdot\text{cm}$$

Tal y como se puede observar, y suponiendo un peso de 70 g por servomotor, el par máximo en estático es de 9.05 kg·cm y en dinámico de 14.6 kg·cm. El servomotor más usado por el resto de los desarrolladores de proyectos similares es el MG996R, cuyo par máximo es 15 kg·cm operando a 6 V y 13 kg·cm a 4.8 V. Por tanto, con esos mismos motores se podría llegar a realizar una implementación en estático, ya que tendríamos un coeficiente de seguridad de 1,65 y en dinámico estaríamos muy justos con 1,02.

Con esta información y teniendo en cuenta que la mayor parte de las implementaciones de marcha suelen tener 3 puntos de apoyo, se apostó por estos servomotores más económicos y testados para esta primera fase, dejando abierta la posibilidad de una mejora en la segunda fase tras emplear el estudio con la herramienta *CAD*.

Por último, tendríamos el resto de los **componentes necesarios para el montaje**:

- Cableado.
- Conectores Dupont para cableado electrónico.
- Cinta termoretráctil.
- Soldador de estaño.
- Estaño para soldadura.
- Herramientas para tratar cables.
- Tornillos M2 y M3 de varios tamaños.
- Tuercas M2 y M3.
- Bridas.
- Protector de cables.
- Cinta americana.
- Extensores de cables para servomotores de 10cm.
- Reglas y pie de rey.
- Impresora 3D.
- Filamento PLA y TPU de varios colores.
- 2 cables USB micro.
- Cargador de batería 2S.
- Alargador de carga de 2S.

#### 5.1.5 Elaboración del esquema electrónico

Una vez seleccionados los componentes, podemos pasar al desarrollo de un primer esquema de conexiones electrónicas empleando la herramienta de desarrollo Fritzing [58]. Esta plataforma cuenta con una extensa librería de componentes electrónicos y permite incorporarlos de forma intuitiva en un desarrollo. La aplicación se encuentra aún en desarrollo y tiene unos cuantos fallos y problemas, pero los resultados son buenos. A continuación, en la Figura 49 se puede ver una primera implementación de este diagrama a excepción de las conexiones para los servomotores. La leyenda de colores para los cables es la siguiente:

- Negro y marrón: tierra o GND.
- Cian: polo positivo de la batería 8.4V.
- Azul oscuro: polo positivo de 6.5V.
- Violeta: polo positivo de 5V.

- 

### 5.1.6 Elaboración de los archivos de impresión

88

estructura de apoyo debido al espesor de las piezas. Además, se obtendrá un resultado similar al molde de inyección y con la mejor resistencia posible sin importar la orientación de impresión. De este modo se evita la subida tan elevada de peso por tan poco incremento de resistencia, a la vez que se mejora los tiempos y el coste.

Se recomienda, por tanto, para esta implementación, un patrón tipo giroide que ofrece las mejores características técnicas para la resistencia sin importar el tipo de esfuerzo al que se vean sometidas las piezas y mitigará la orientación de impresión. Los perfiles o configuración para la impresión estarán disponibles tanto para PLA como para TPU en la [carpeta compartida](#).

El proceso una vez realizado esto consiste en abrir, colocar las piezas para la impresión y segmentar el modelo en capas. Una vez realizada la segmentación (Figura 50), se obtendrá un archivo de movimientos llamado “GCODE”. Este no es más que una lista de comandos cartesianos de movimiento para cada motor de la impresora, esta imprimirá capa a capa nuestros modelos. Un ejemplo de la vista en la aplicación es la siguiente:

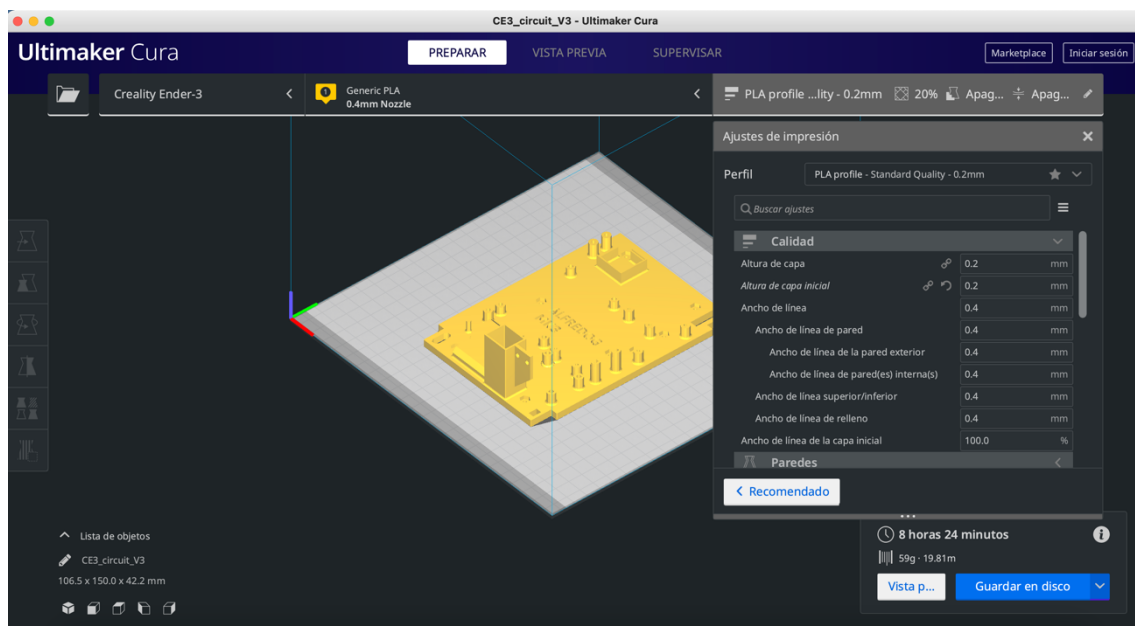


Figura 50. Vista tras la segmentación de la placa del circuito de la fase 3 (Fuente propia).

En la esquina inferior derecha están los costes temporales y de material. Con estos datos y los de el punto anterior, se podrán elaborar las tablas de costes del 5.1.7. Por último, se cargarán estos archivos a la memoria de la impresora y estarán listos para imprimir.

#### 5.4.4 Elaboración de las tablas de costes

A continuación, tras recopilar la información de los componentes necesarios y de las impresiones, se obtienen los costes totales. Finalmente se ha escogido el PLA ya que no produce gases dañinos durante su impresión en el interior y es más sencillo de imprimir. También se ha optado por TPU para las patas de “goma” con el fin de absorber mejor el impacto y ofrecer un mejor agarre. La tabla 1 se encuentra a continuación:

PROTOTIPO – FASE 1 – COSTES DE IMPRESIÓN									
--	--	--	--	--	--	--	--	--	--

Hombros y soportes									
Cantidad	Nombre de la pieza		Relleno	Duración	Filamento		Total de la duración	Total filamento	de
2x	Inner Shoulder		20%	1h 29min	16	g	2h 58min	32	g
2x	Outer Shoulder		20%	1h 33min	16	g	3h 6min	32	g
2x	Bottom Shoulder		20%	36min	6	g	1h 12min	12	g
2x	Top Shoulder		20%	54min	10	g	1h 48min	20	g
2x	Chassis Side		20%	1h 7min	11	g	2h 14min	22	g
1x	Circuitboard Mount		20%	2h 42min	30	g	2h 42min	30	g
4x	Shoulder Joint Servomount		20%	1h 10min	12	g	4h 40min	48	g
4x	Shoulder Joint Servohorn		20%	41min	7	g	2h 44min	28	g
4x	Cap								
4x	Shoulder Joint Ball Bearing		20%	5min	0	g	20min	2	g
4x	Pin								
							Total	226	g

Miembros									
Cantidad	Nombre de la pieza		Relleno	Duración	Filamento		Total de la duración	Total filamento	de
4x	Limb_Bottom_Shell		20%	2h 10min	25	g	8h 40min	100	g
4x	Limb_Top_Shell		20%	2h 7min	23	g	8h 28min	92	g
4x	Limb_Ball_Bearing_Mount		20%	29min	5	g	1h 56min	20	g
4x	Limb_Servohorn_Mount		20%	31min	5	g	2h 4min	20	g
							Total	232	g

Pies									
Cantidad	Nombre de la pieza		Relleno	Duración	Filamento		Total de la duración	Total filamento	de
4x	Wrist		20%	2h 1min	26	g	8h 4min	104	g
4x	Wrist Counterpart		20%	47min	9	g	3h 8min	36	g



4x	Shoulder Joint Ball Bearing Pin	20%	5min	0	g	20min	2	g
4x	Foot Tip	20%	51min	6	g	51min	24	g
						Total	166	g

Cubiertas del chasis y circuito								
Cantidad	Nombre de la pieza		Relleno	Duración	Filamento	Total de la duración	Total de filamento	
1x	Front_Cover		100%	6h 2min	63 g	6h 2min	63	g
1x	Rear_Cover		100%	4h 37min	53 g	4h 37min	53	g
1x	Top_Cover_Split_Front		100%	4h 3min	48 g	4h 3min	48	g
1x	Top_Cover_Split_Rear		100%	2h 21min	28 g	2h 21min	28	g
1x	Bottom_Cover_Split_Front		100%	3h 53min	46 g	3h 53min	46	g
1x	Bottom_Cover_Split_Rear		100%	2h 14min	26 g	2h 14min	26	g
1x	Circuitboard_mount		20%	8h 14min	65 g	8h 14min	65	g
						Total	329	g

Total PLA Negro	368 g
Total PLA Amarillo	561 g
Total TPU Blanco	24 g
Total de filamento	953 g

Tabla 1. Costes de impresión de la fase 1 (Fuente propia).

En cuanto a la tabla de costes con todos los materiales y el equipo necesario para llevar esta fase a cabo, se puede ver a continuación:

Nombre	Modelo	Cantidad	Precio €	Total
Sevomotores	MG996R Servo 12Kg	12	5	60
Microcontrolador	ESP32-DevKitC	1	9	9
Cámara	ESP32-CAM	1	10	10
Conexión TTL	FT232RL USB-TTL-Adapter	1	8	8
Protector de cables	3m de malla de cable	1	8	8
Protector de cables	Tubos termoretráctiles	1	15	15
Extensor de cables	Servocable Extension	4	10	40
Rodamientos	625ZZ Ball Bearing	8	1	8
Luces	WS2812b 12 LED Ring (50mm outer diameter)	1	8	8
Sensores de distancia	HC-SR04 Ultrasonic Sensor	2	3	6
Sensor inercial	GY-521 Gyroscope and Accelerometer	1	4	4
Controladora de motores	PCA9685 16Channel 12Bit PWM Board	1	14	14

Relé	HW-482 / KY-019 5V 10A Relais	1	8	8
Interruptor	19mm Push Button with LED	1	10	10
Conexión externa USB	Micro-USB to DIP Adapter	2	6	12
Transformador principal	XL4016 DC-DC 9A Stepdown Converter	1	14	14
Transformador secundario	LM2596 DC-DC Stepdown Module	1	10	10
Batería de litio	6200mAh LiPo 80C 2S	1	35	35
Conectores Dupont	Pack de conectores dupont con herramienta crimpadora	1	30	30
Cables 16AWG	Cables 16AWG 20m	1	20	20
Pack de destornillador	Destornillador con cabezales	1	15	15
Pantalla	1,8" TFT with ST7735 w/o SD	1	14	14
Soldador	Soldador de estaño con estaño	1	20	20
Bridas	Bridas de 20cm	1	5	5
Tornillos	M2x8 cylinderhead screws + M2 nuts (100)	1	8	8
Tornillos	M3x8 cylinderhead screws + M3 nuts (100)	2	8	16
Tornillos	M3x20 cylinderhead screws + M3 nuts (100)	1	12	12
Cables, conectores y otros	Dupont bundle	1	20	20
PLA	PLA 1 kg	2	20	40
TPU	TPU 500 g	1	20	20
Amoladora	Amoladora con herramientas para pulido y taladrado	1	30	30
Pelador de cables	Pelador de cables	1	15	15
Impresora 3D	Impresora 3d Ender 3 de creality	1	250	250
Conectores T	Pack de conectores T	1	8	8
Terminales para 16AWG	Pack de terminales 16AWG	1	4	4
Pistola de silicona	Pistola de silicona con silicona	1	20	20
Cargador de batería	Cargador 2s más alargador	1	30	30
Cinta para cables	Pack de colores de cinta para cables	1	5	5
Total	No incluye el gasto de luz			856

Tabla 2. Costes de materiales y equipo de la fase 1 (Fuente propia).

Se puede observar como el coste de desarrollo del prototipo en esta fase es relativamente bajo, aunque muy superior a las expectativas iniciales. Los precios son dependientes del mercado en ese momento, del lugar de compra y los tiempos, por tanto, el coste de replica puede ser distinto del proporcionado.

#### 5.1.8 Impresión de las piezas estructurales, problemas y soluciones

Para la impresión de las piezas, se empleó una impresora 3D modelo *Ender 3*. Tras realizar la calibración de ejes y de cama, cargar el plástico, se procedió con la impresión de todas las piezas con una duración total de 88 h divididas en un total de 26 días debido a los solapes, problemas técnicos y enfriamientos necesarios entre impresiones.

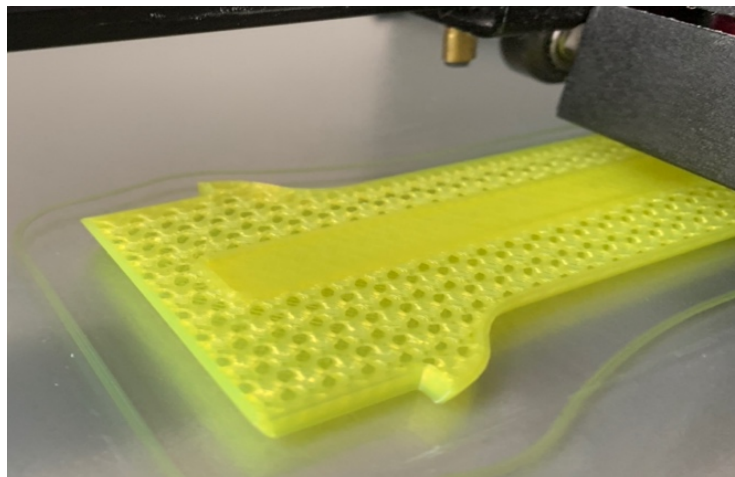


Figura 51. Impresión de un miembro superior con vista del patrón giroide (Fuente propia).

Durante el transcurso de las numerosas impresiones a lo largo y ancho de las distintas fases, se han encontrado problemas como nudos en el filamento, descalibrado de la cama caliente, infra y sobre extrusión de piezas resultando en el fenómeno del pie de elefante, desplazamiento de capas a media impresión, plástico defectuoso e incluso atascamiento explosivo del extrusor. Por y gracias a estos problemas, he aprendido mucho sobre el funcionamiento de impresoras 3D hasta el punto de desarrollar mi propio sistema de calibrado automático con compensación bilineal corriendo en un Marlin 2.0 personalizado e instalado ya en varias impresoras 3D. Esta información se encuentra expandida en el [anexo tercero](#).

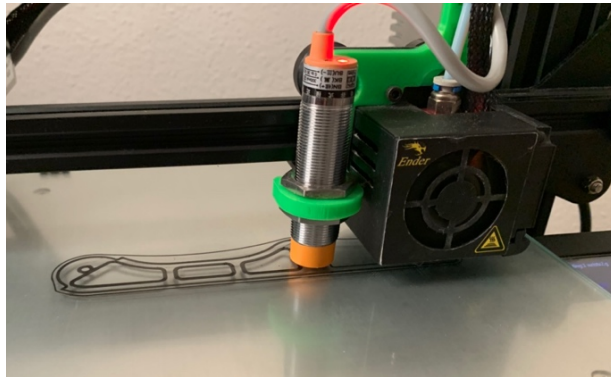


Figura 52. Impresión de una pata inferior de la fase 3 con el sistema de calibrado y nivelación automática (Fuente propia).

## 5.2 Construcción y pruebas básicas: Fase 1

Con el objetivo de no extender la memoria, se ha realizado un [vídeo explicativo](#) de cada una de las fases en las que se desarrolla la preparación y construcción del prototipo además de una secuencia a cámara rápida de su desmontaje. En este vídeo se detallarán algunos de los problemas encontrados y se podrán visualizar imágenes u vídeos de pruebas para todos los componentes en distintas fases de implementación. Por ende, en los puntos 5.2, 5.4 y 5.6, se incluirán algunas fotografías o montajes de estas a modo de ilustración del proceso, y se describirá mayoritariamente de forma escrita. Además, debido a la planificación por fases, se encontrarán más imágenes del desarrollo de la construcción y verificación de manera progresiva ya que hay muchos componentes comunes entre las mismas susceptibles de modificaciones. Por esta razón el aporte multimedia será mayor en la fase tercera por ser la más relevante del proyecto.

### 5.2.1 Preparación del cableado eléctrico para el circuito principal y montaje

El primer paso de esta fase de construcción consistió en preparar el cableado necesario para ensamblar el circuito eléctrico. Para ello se instalaron los sensores en sus respectivas posiciones sobre la plantilla del circuito (Figura 53) y midiendo las distancias y trazas aproximadas de los cables. Se comprobó además que la entrada de la placa de control de los servomotores no era lo suficientemente grande para albergar el cable de 16 AWG escogido para llevar la alta corriente necesaria para los servomotores. Por esta razón, se substituyó posteriormente por otro terminal de conexiones de mayor tamaño que necesitó un cable intermedio debido al menor espacio para su montura.

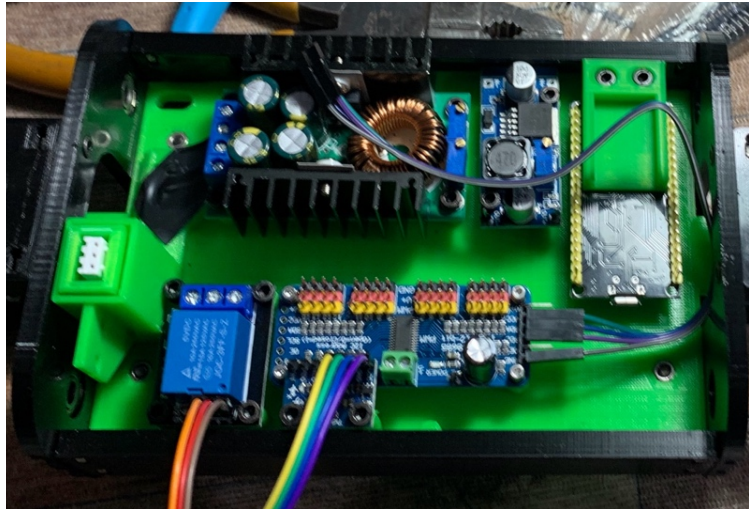


Figura 53. Componentes sobre el circuito de la fase 1 (Fuente propia).

Realizado este, se procedió a la conexión del resto de componentes del prototipo y su ensamblaje ([ver vídeo del montaje para más detalle](#)). Resultando en una primera versión (Figura 54) de la implementación visible en la siguiente imagen:



Figura 54. Prototipo de la fase 1 (Fuente propia).

### 5.2.2 Elaboración de códigos de testeo y evaluación del movimiento de las articulaciones

Una vez obtenido el primer prototipo, se consideró necesario evaluar su rendimiento mediante una serie de pruebas básicas. Para ello, se elaboró una serie de códigos en Arduino referenciados en el [anexo 1](#), que comprobasen los distintos sensores y componentes del equipo. En primer lugar, se realizaron pruebas sobre los sensores de ultrasonidos, este consiste en una ecuación basada el tiempo de vuelo y la constante de la velocidad del sonido. Seguidamente se comprobó la cámara con el código de ejemplo de servidor *CAM* proporcionado por la librería de la ESP. Este código consistió en establecer un servidor en la red local en el que se realiza una retransmisión de vídeo. En lo referente al relé, bastaba con un simple comando de bit alto o bajo. Tras esta prueba, se comprobó que el relé era defectuoso y se anotó para ser reemplazado en la segunda fase. La pantalla *TFT* funcionaba correctamente con el código del fabricante, pero se observó que al requerir seis entradas *GPIO* de la ESP32, la posibilidad de mostrar la cámara y más información no compensaba la pérdida de puertos para la expansión sensorial necesaria. Por este motivo se optó por su reemplazo por un modelo distinto en la segunda fase.

Finalmente, llegó el punto de la prueba de movimiento de los servomotores. Para ello se desarrolló un código que empleaba un joystick con 2 grados de libertad asociados a las distintas posiciones de los servomotores para comprobar los movimientos. El resultado de esta prueba fue que se observó que el movimiento entre las distintas posiciones era posible, pero con cierta dificultad y backlash. Al probar esos códigos con todas las articulaciones a la vez, el robot presentó dificultades para mantenerse en las posiciones indicadas. Tras desensamblarlos se comprobó que, aunque funcionaban todos tal y como se verificó tras la compra salvo por un factor importante, no todos tenían la misma potencia, velocidad, ángulos de llegada e incluso algunos de ellos presentaban saltos entre posiciones. De hecho, como consecuencia de esto, uno de los puntos de sujeción de los servomotores se partió a torsión como se puede ver en la Figura 55. Es por ello por lo que se optaría para la segunda fase en realizar un cambio de los servomotores por unos con mejores prestaciones y con mayor fiabilidad.

Como conclusión de esta primera fase y tras las pruebas realizadas, se verificó que efectivamente era posible continuar con la implementación del prototipo, se comprobó la validez del modelo y se detectaron cambios necesarios



en la pantalla, relé y servomotores que deberían de ser modificados y sustituidos en la siguiente fase.



Figura 55. Rotura a torsión del Servohorn delantero izquierdo (Fuente propia).

## 5.3 Diseño e impresión del prototipo: Fase 2

### 5.3.1 Planteamiento de mejoras al sistema

Tras el estudio del sistema inicial, se plantearon algunas mejoras, en función de los defectos y carencias encontrados en el sistema estándar de la Fase 1. Estas mejoras implican la selección de una actuación más apropiada. Para ello, se realizará una simulación de movimiento dinámico y de esfuerzos y una simulación de las tensiones sobre el modelo. Para ello se utilizará la herramienta *CAD/CAE Autodesk Inventor*, que proporcionará los valores necesarios para el diseño y la elección.

En cuanto a la pantalla, se buscará una que preferiblemente emplee el menos número conexiones. Es decir, que funcione empleando el protocolo de comunicación I2C. También será necesario reevaluar si el sistema de alimentación tras la modificación en el sistema de actuación requerirá cambios, así como las piezas estructurales asociadas. Esto implica modificar el diseño de las piezas de soporte en función de los nuevos componentes seleccionados y

que hagan uso de estas piezas. En este caso, también habrá que rediseñar el cableado y diagrama de conexiones.

Adicionalmente, se pretende explorar la posibilidad de explorar un *RADAR/LiDAR* con sensores de ultrasonidos, que se monte en la parte superior del robot y le permita obtener información más precisa del entorno. Este sistema también incluirá un anillo LED para transmitir directamente información visual. El diseño de este sistema se realizará, de nuevo con la herramienta *CAD/CAE* Autodesk Inventor.

### 5.3.2 Estudio dinámico del sistema con CAD/CAE

Para seleccionar el correcto accionamiento, se ha podido comprobar que, empleando las ecuaciones de carga estática y dinámica, no se obtiene el resultado deseado a pesar de que el cálculo indicaba un correcto funcionamiento. Es por ello por lo que, tras incorporar los parámetros de la cinemática del robot al modelo ensamblado de una pierna y los valores correspondientes a sus características físicas, se ha podido realizar una simulación del movimiento, de la que se pueden extraer los momentos y pares de inercia necesarios para cualquier movimiento como ilustra la Figura 56.

Tras la configuración del ensamblaje en Autodesk Inventor, se cargó el ensamblaje "*LIMB\_FRONT\_LEFT.iam*" y se le programó una trayectoria a cada uno de sus grados de libertad. El objetivo era mantener un movimiento con una velocidad constante de 20 °/s con el objetivo de dorar de uniformidad los datos y simular un movimiento de marcha. Siguiendo esta estrategia y teniendo en cuenta que la articulación del hombro flexor/extensor (*shoulder joint*) tiene un rango máximo de movimiento de  $\pm 30^\circ$ , se programó un movimiento constante alternando en su rango de movimiento cada 1.5 s (Figura 57). En el caso del hombro abductor (*limb joint*) y del codo (*wrist joint*) (Figura 58), fue una trayectoria de  $180^\circ$  con un valle manteniendo la flexión completa 1.5 s y el movimiento inverso. Se puede visualizar la configuración, así como las trayectorias introducidas al modelo, en las figuras 57 y 58 disponibles a continuación:



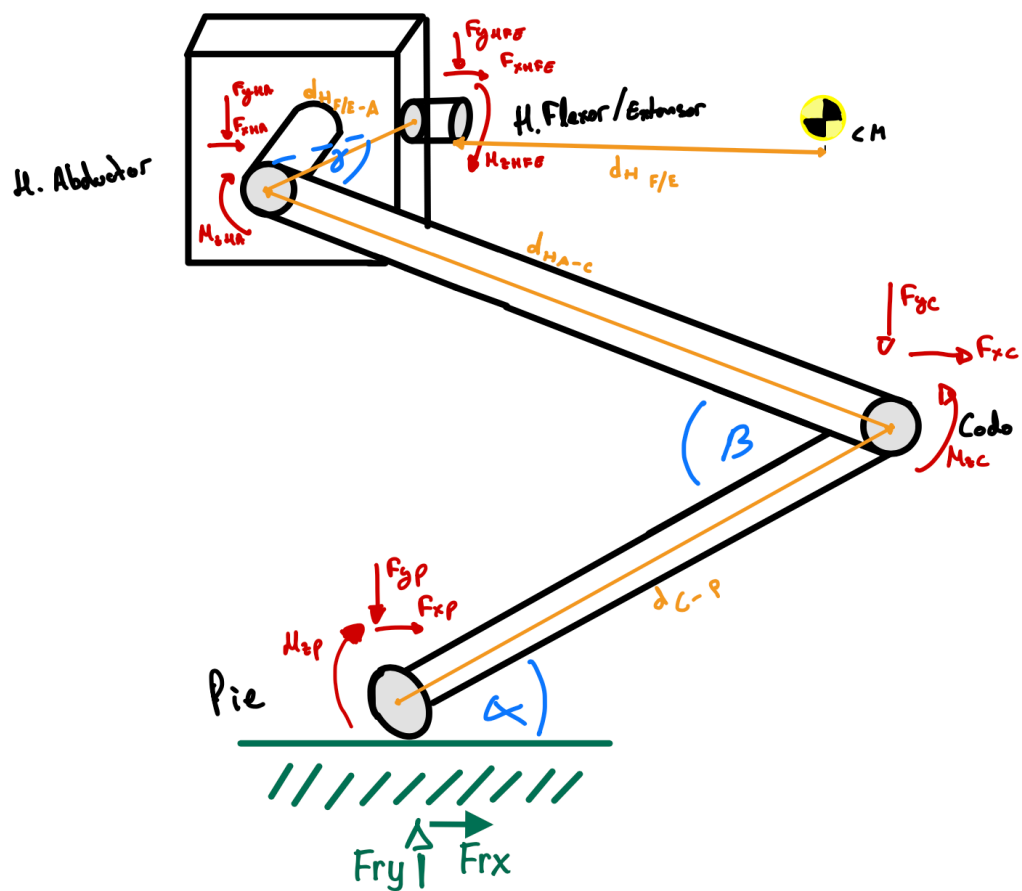


Figura 56. Ilustración de las medidas deseadas (Fuente propia).

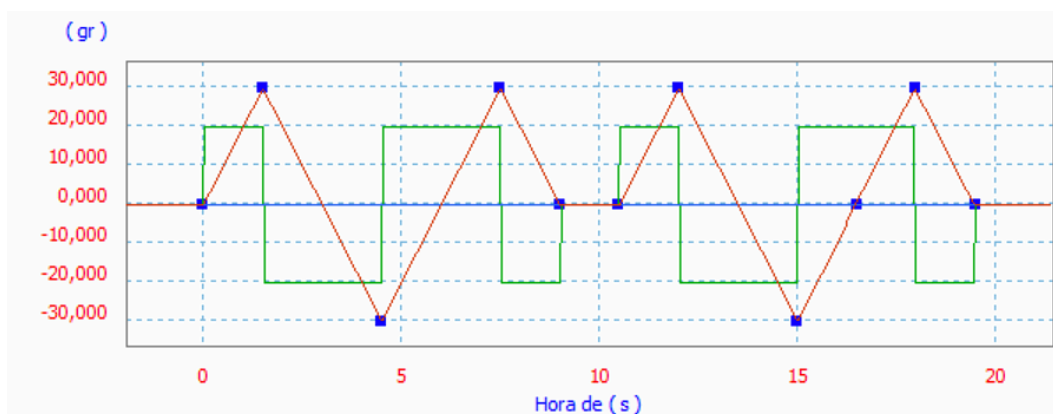


Figura 57. Movimiento impuesto al grado de libertad correspondiente al hombro flexor/extensor (Fuente propia).

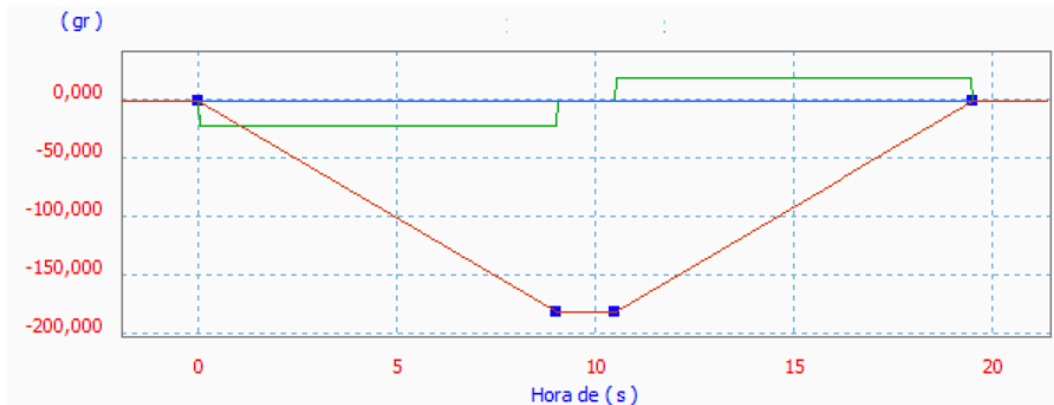


Figura 58. Movimiento impuesto al grado de libertad correspondiente al hombro abductor y al codo (Fuente propia).

Una vez impuestos los movimientos, se procedió a la simulación dinámica. El desarrollo y la implementación están disponibles en un vídeo en [la carpeta en la nube](#), en la que se puede apreciar el resultado de la simulación fotograma a fotograma y acelerado 50 veces (aproximadamente un 50% del tiempo real), como se puede ver en la siguiente imagen (Figura 59):

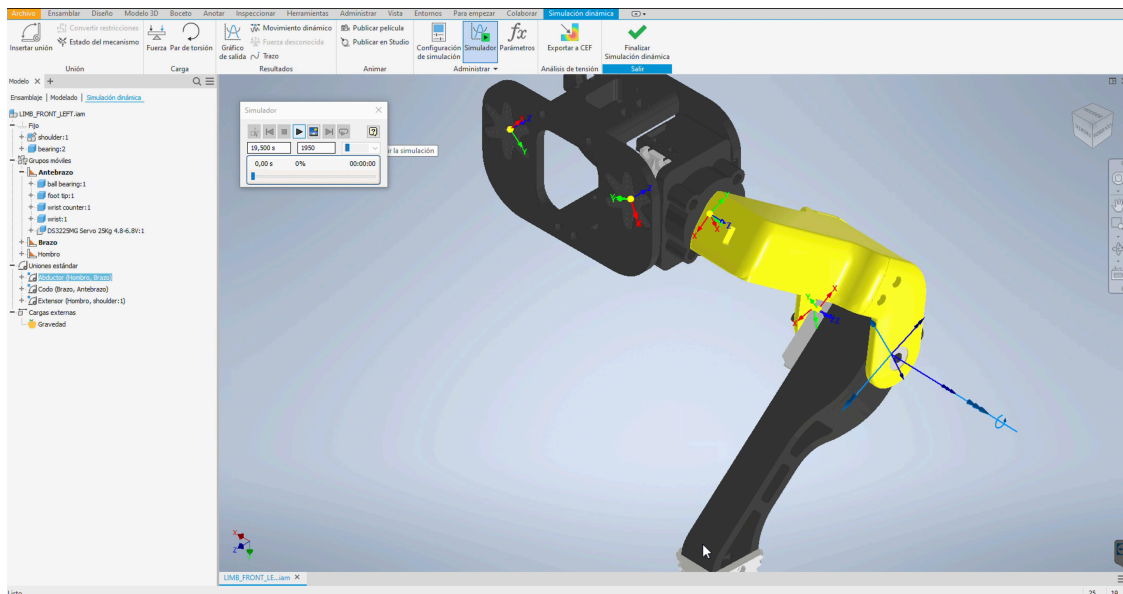


Figura 59. Configuración del análisis dinámico instantes antes de su ejecución (Fuente propia).

Cuarenta minutos después, el equivalente a 1 paso por segundo de un total de 1952 pasos o un paso por centésima de segundo, se obtuvieron los resultados de la simulación dinámica. Los enlaces a cada una de las tablas se encuentran en el [Anexo 4º](#) debido a su extensión (155 páginas por tabla).

Una vez computados esos números, se crearon 3 gráficas (Figuras 60 a 62) con los resultados obteniendo los valores para los pares máximos para un movimiento sin contacto y con una carga de la propia articulación. Seguidamente se repitió el proceso aplicando una carga de 30 N en la base del soporte de los hombros (visible en la Figura 63 el caso del abductor ya que el único cambio es la línea que representa la fuerza de 30 N añadida) para evaluar el cambio entre el movimiento con y sin carga. Los resultados obtenidos fueron los siguientes:

<i>Simulación dinámica en valor absoluto</i>	<i>Fuerza máxima (N)</i>	<i>Par máximo (N·mm)</i>
<i>Sin carga</i>	4.78	451.46
<i>Con carga de 30N</i>	4.78	451.46

Tabla 3. Comparativa de resultados de la simulación dinámica (Fuente propia).

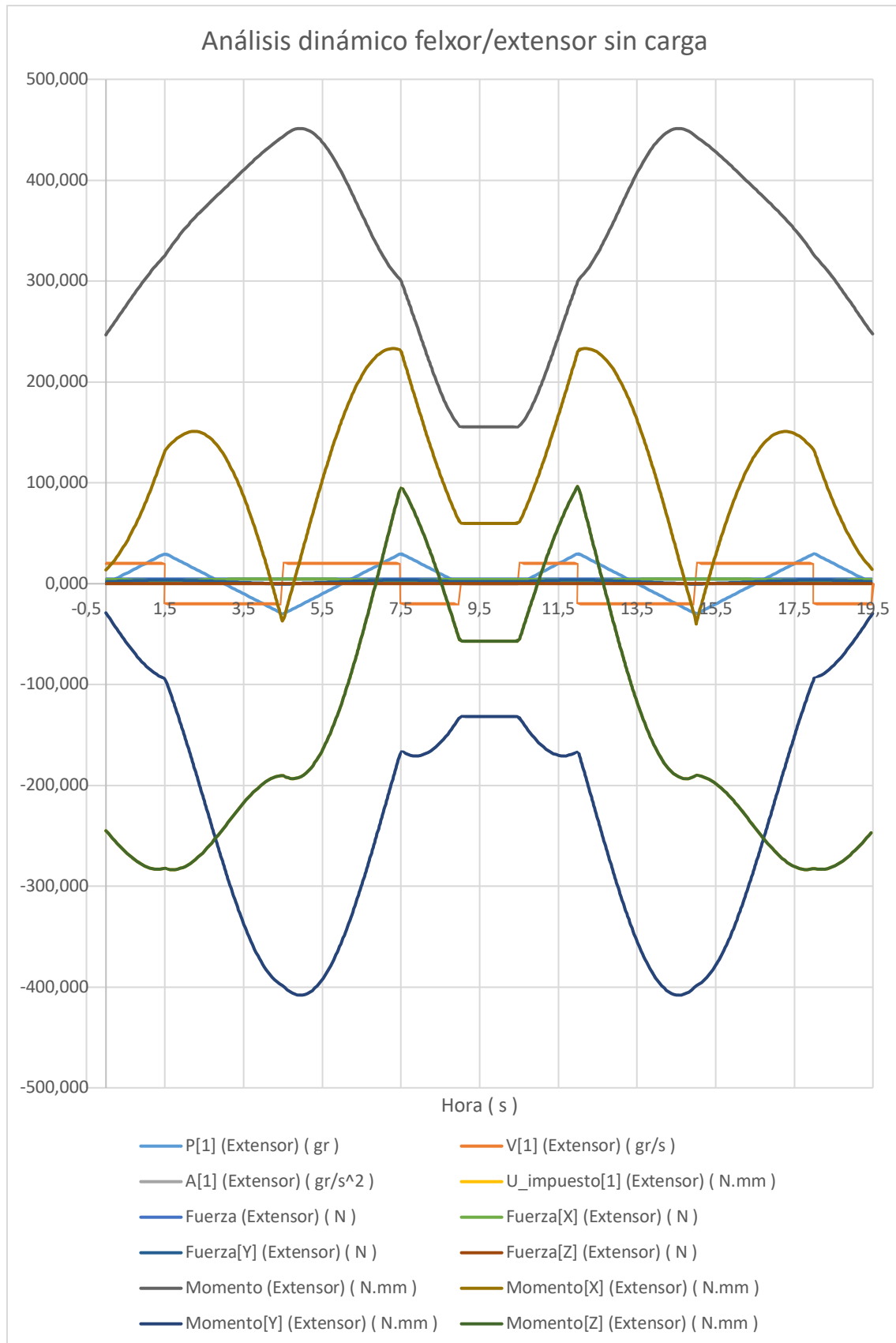


Figura 60. Resultados del análisis dinámico para el hombro flexor/extensor (Fuente propia).

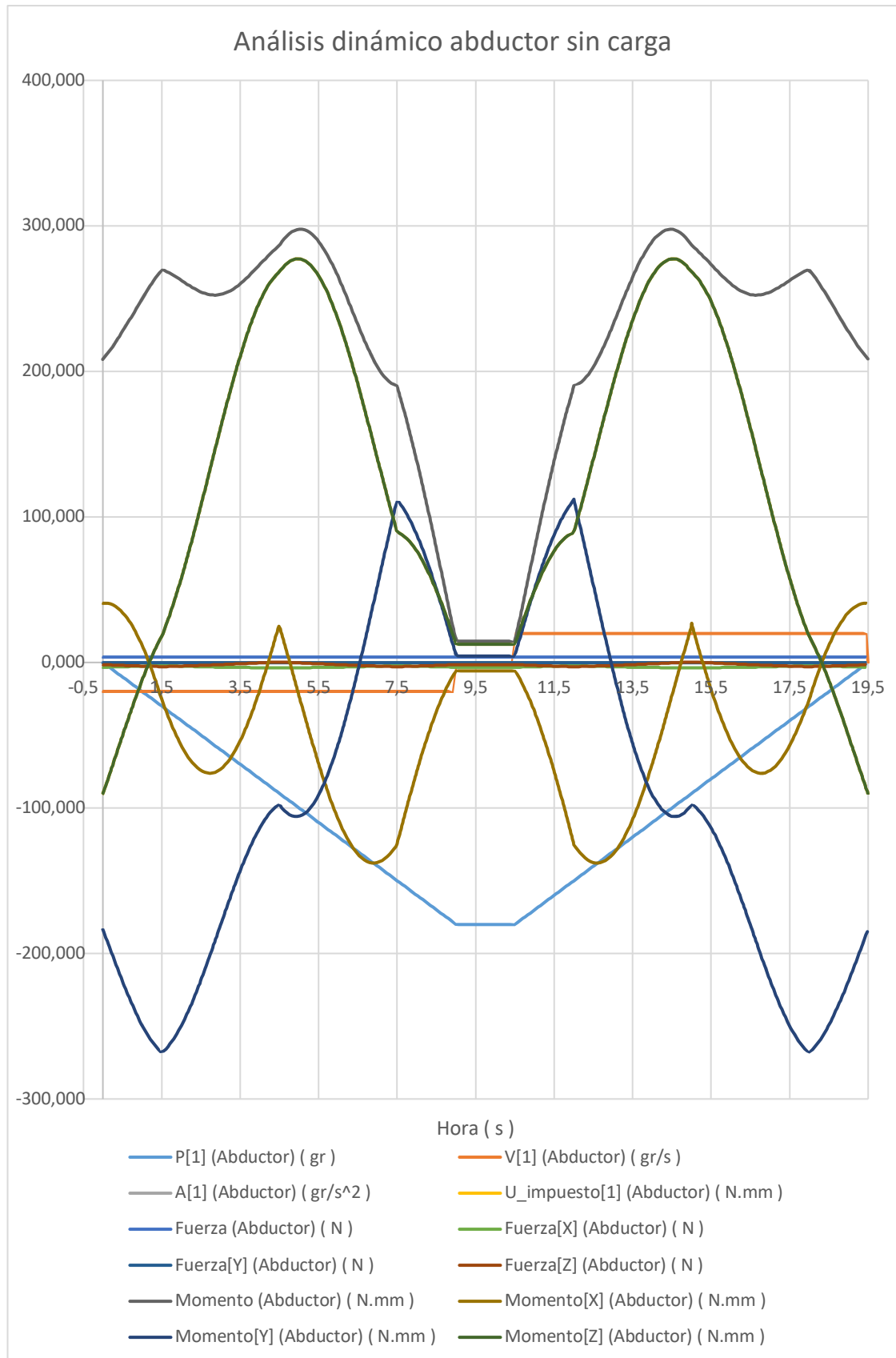


Figura 61. Resultados del análisis dinámico para el hombro abductor (Fuente propia).

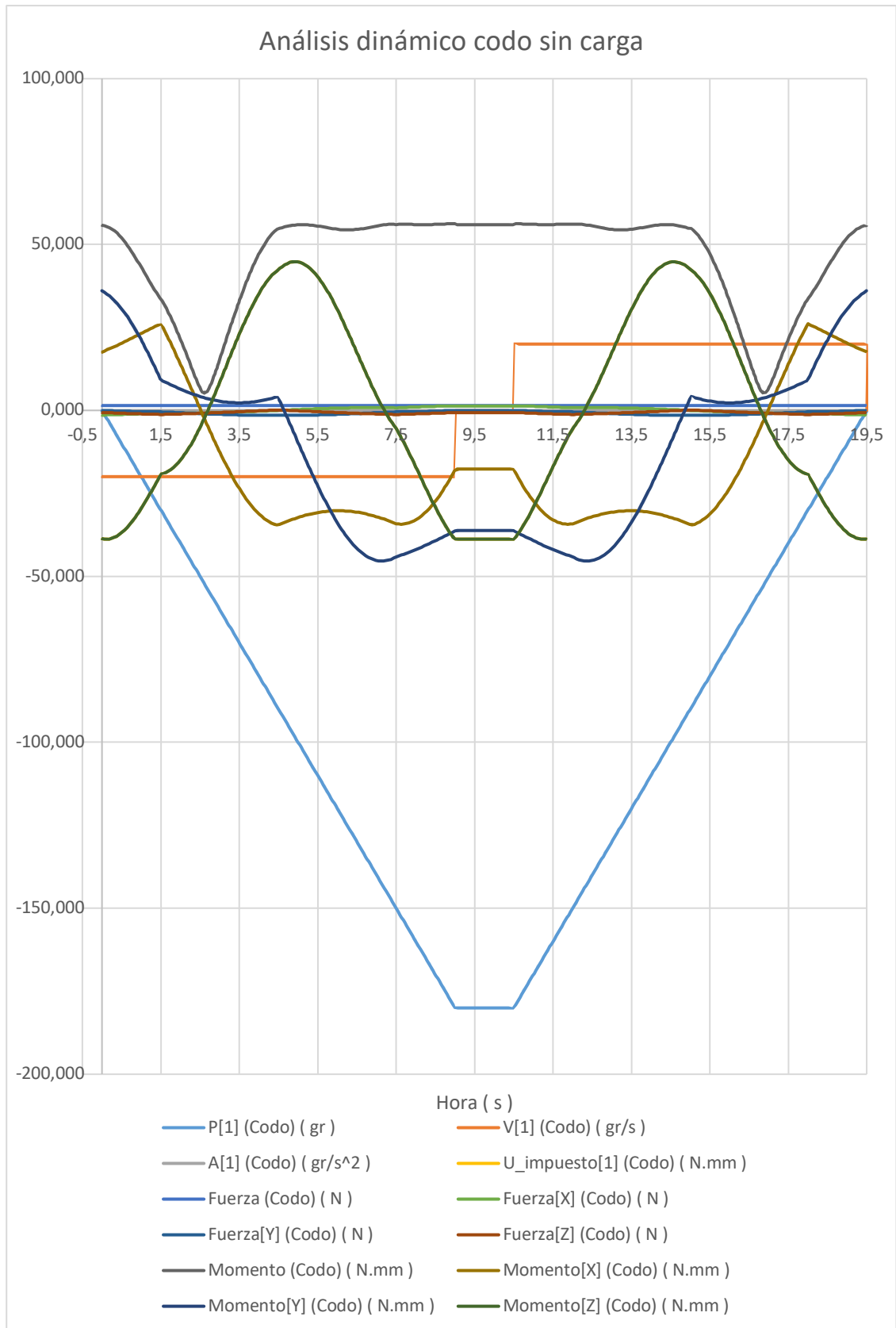


Figura 62. Resultados del análisis dinámico para el hombro abductor (Fuente propia).

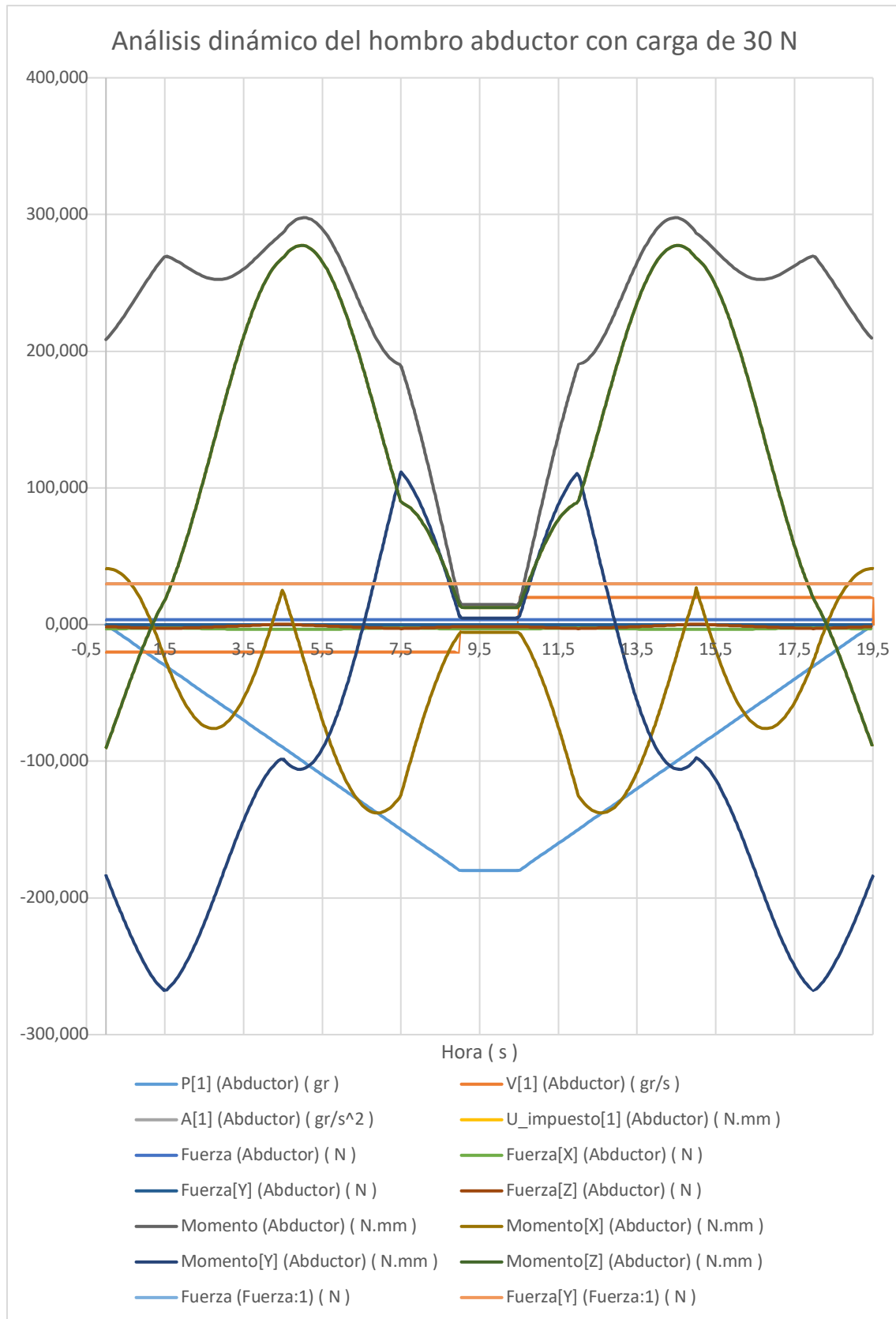


Figura 63. Resultados del análisis dinámico para el hombro abductor con una carga de 30N (Fuente propia).

De los datos extraídos en la tercera tabla, se concluyó que el nuevo servomotor debería de tener al menos un par de 4.6 kg·cm para garantizar el movimiento sin apoyo dinámico. Teniendo esto en cuenta, se volverán a realizar los cálculos manuales para un factor de seguridad de 1.5 suponiendo un peso de 60 g por servo motor. Posteriormente se escogerá el accionamiento más adecuado en función del valor más restrictivo.

- $X_n$  = peso del sistema hasta el punto en la cadena cinemática donde se acopla el servomotor.
- $y$  = peso del cuerpo del robot = 1.9 kg.

$$M_{SHA} = (y)/4 \cdot \text{sqrt}((20.75/2)^2 + (7.8/2)^2) = 5.26 \text{ kg}\cdot\text{cm}$$

$$M_{SHFE} = (y/4 + x_1 + p) \cdot 6.05 = (1.9/4 + 0.054 + 0.06) \cdot 3.525 = 2.076 \text{ kg}\cdot\text{cm}$$

$$M_{SC} = (y/4 + x_2 + p) \cdot 11.12 = (1.9/4 + 0.120 + 0.054 + 0.12) \cdot 11.12 = 8.55 \text{ kg}\cdot\text{cm}$$

Conocidos estos datos, se hará lo mismo para calcular el par en estático para 2 puntos de apoyo:

$$M_{SHA} = (y)/2 \cdot \text{sqrt}((20.75/2)^2 + (7.8/2)^2) = 11.08 \text{ kg}\cdot\text{cm}$$

$$M_{SHFE} = (y/2 + x_1 + p) \cdot 6.05 = (1.9/2 + 0.054 + 0.06) \cdot 3.525 = 3.75 \text{ kg}\cdot\text{cm}$$

$$M_{SC} = (y/2 + x_2 + p) \cdot 11.12 = (1.9/2 + 0.120 + 0.054 + 0.12) \cdot 11.12 = 13.83 \text{ kg}\cdot\text{cm}$$

Multiplicando estos por el factor de seguridad se obtiene que el par requerido por la articulación con mayor esfuerzo es de 20.48 kg·cm. Por tanto, se deberá hallar un servomotor que satisfaga o mejore estas condiciones ya que son más elevadas que las obtenidas en ambas simulaciones dinámicas. Esto puede deberse a la simplicidad de la simulación y podría requerir una simulación más completa con el resto del modelo y un terreno plano inferior para brindarnos resultados más aproximados a la realidad en el caso de buscar el par exacto en apoyo. No obstante, y en base a la experiencia adquirida, los datos son acertados y se aproximan bastante a lo visto en la experimentación con el robot real. Por tanto, y ya que se ha aplicado un factor de seguridad para acomodar las pequeñas variaciones y errores acumulados en el cálculo, se tomarán como suficientemente válidos.

### 5.3.3 Estudio de tensiones del sistema con CAD/CAE

Del mismo modo, aprovechando el ensamblaje de una de las extremidades, se ha decidido someter a una carga de 30 N, que es similar al peso final del robot, para realizar un estudio de tensiones y deformaciones. Esta carga se ha colocado sobre el eje de carga del servomotor y sobre el núcleo



central de la estructura de carga de los hombros. Si bien es cierto que la carga es soportada realmente por 4 extremidades, de esta forma podríamos conocer la capacidad de carga requerida por los servomotores en el caso de que solo haya un punto de apoyo predominante. Este estudio también permite obtener el límite estructural de los componentes ya que se observó que el servohorn rompió a torsión en una circunstancia similar. Esto permitirá conocer los límites del robot y evaluar un posible refuerzo de algún componente.

A continuación, se puede observar unos ejemplos visuales (figuras 64 y 65) de los resultados obtenidos, así como ambas tablas (4 y 5) con los datos, para estos análisis de tensiones, hallándose los enlaces a los informes completos en el [Anexo 5](#):

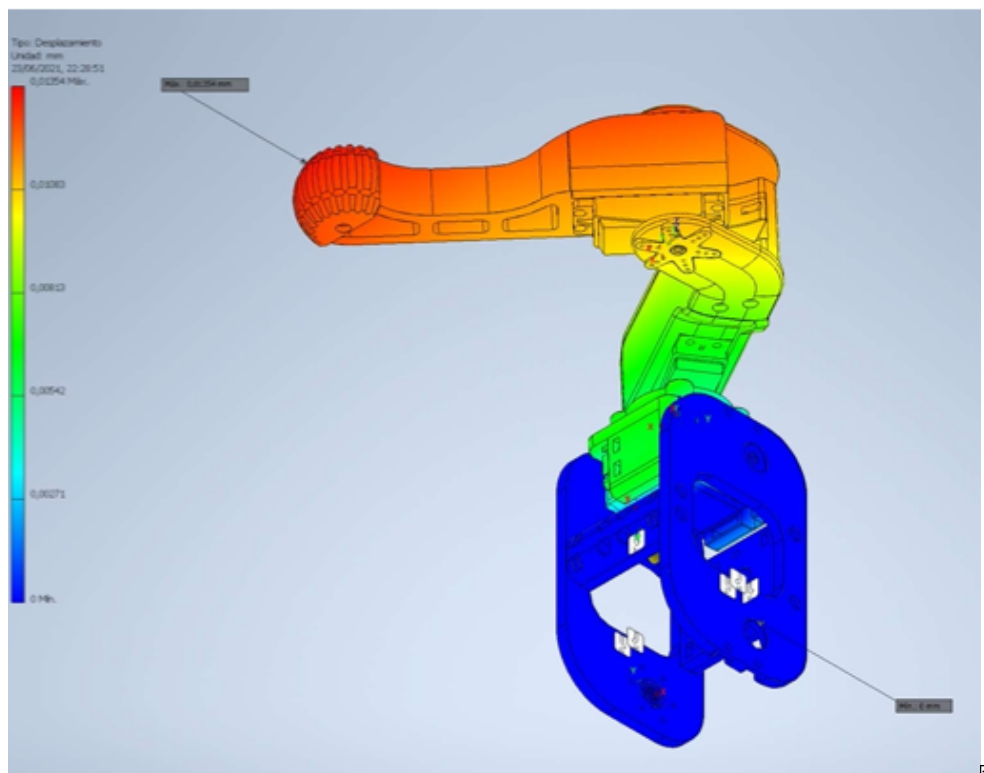


Figura 64. Ejemplo 1 de los resultados del análisis estático de tensiones distribuidas (Fuente propia).

<b>Nombre</b>	<b>Mínimo</b>	<b>Máximo</b>
Volumen	374751 mm <sup>3</sup>	
Masa	0,57143 kg	
Tensión de Von Mises	0,0000000112725 MPa	30,8091 MPa
Primera tensión principal	-4,37231 MPa	17,1692 MPa
Tercera tensión principal	-24,1681 MPa	4,24219 MPa
Desplazamiento	0 mm	0,0135431 mm
Coeficiente de seguridad	1,49768 su	15 su
Tensión XX	-12,1456 MPa	11,6612 MPa
Tensión XY	-12,0325 MPa	12,6642 MPa
Tensión XZ	-9,39228 MPa	9,4529 MPa
Tensión YY	-18,976 MPa	13,4831 MPa
Tensión YZ	-7,07558 MPa	11,4581 MPa
Tensión ZZ	-6,94539 MPa	6,07587 MPa
Desplazamiento X	-0,00388582 mm	0,00800179 mm
Desplazamiento Y	-0,01207 mm	0,00303035 mm
Desplazamiento Z	-0,00324173 mm	0,000254209 mm
Deformación equivalente	0,000000000052033 su	0,00563065 su
Primera deformación principal	-0,0000127576 su	0,00348673 su
Tercera deformación principal	-0,00603653 su	0,0000108284 su
Deformación XX	-0,00345904 su	0,00344029 su
Deformación XY	-0,0026948 su	0,0043802 su
Deformación XZ	-0,00252025 su	0,000791147 su
Deformación YY	-0,00378903 su	0,00215866 su
Deformación YZ	-0,000656439 su	0,000748193 su
Deformación ZZ	-0,000593765 su	0,000580139 su
Presión de contacto	0 MPa	23,1325 MPa
Presión de contacto X	-21,8342 MPa	22,8732 MPa
Presión de contacto Y	-11,3696 MPa	9,9855 MPa
Presión de contacto Z	-2,67014 MPa	3,1818 MPa

Tabla 4. Resultados del análisis de tensión con cargas distribuidas de 30N (Fuente propia).

<b>Nombre</b>	<b>Mínimo</b>	<b>Máximo</b>
Volumen	374747 mm <sup>3</sup>	
Masa	0,571425 kg	
Tensión de Von Mises	0,0000141891 MPa	18,7338 MPa
Primera tensión principal	-2,88858 MPa	25,7195 MPa
Tercera tensión principal	-14,5308 MPa	5,10901 MPa
Desplazamiento	0 mm	0,0238877 mm
Coeficiente de seguridad	1,45916 su	15 su
Tensión XX	-6,99788 MPa	8,82705 MPa
Tensión XY	-5,3992 MPa	5,6933 MPa
Tensión XZ	-5,77702 MPa	5,78663 MPa
Tensión YY	-10,7621 MPa	20,5372 MPa
Tensión YZ	-5,94218 MPa	7,61553 MPa
Tensión ZZ	-10,5548 MPa	11,5118 MPa
Desplazamiento X	-0,00571566 mm	0,0213115 mm
Desplazamiento Y	-0,00397578 mm	0,015168 mm
Desplazamiento Z	-0,00975561 mm	0,00232548 mm
Deformación equivalente	0,00000000205069 su	0,00562955 su
Primera deformación principal	-0,00000603774 su	0,00495432 su
Tercera deformación principal	-0,00479422 su	0,00000013625 su
Deformación XX	-0,00105616 su	0,00190498 su
Deformación XY	-0,00254614 su	0,00102567 su
Deformación XZ	-0,00164193 su	0,00356498 su
Deformación YY	-0,00133754 su	0,00107804 su
Deformación YZ	-0,00366081 su	0,0020075 su
Deformación ZZ	-0,000962778 su	0,000958626 su
Presión de contacto	0 MPa	85,6097 MPa
Presión de contacto X	-15,4574 MPa	18,3607 MPa
Presión de contacto Y	-19,1437 MPa	76,5073 MPa
Presión de contacto Z	-11,6161 MPa	35,1672 MPa

Tabla 5. Resultados del análisis de tensión con una única carga de 30N (Fuente propia).

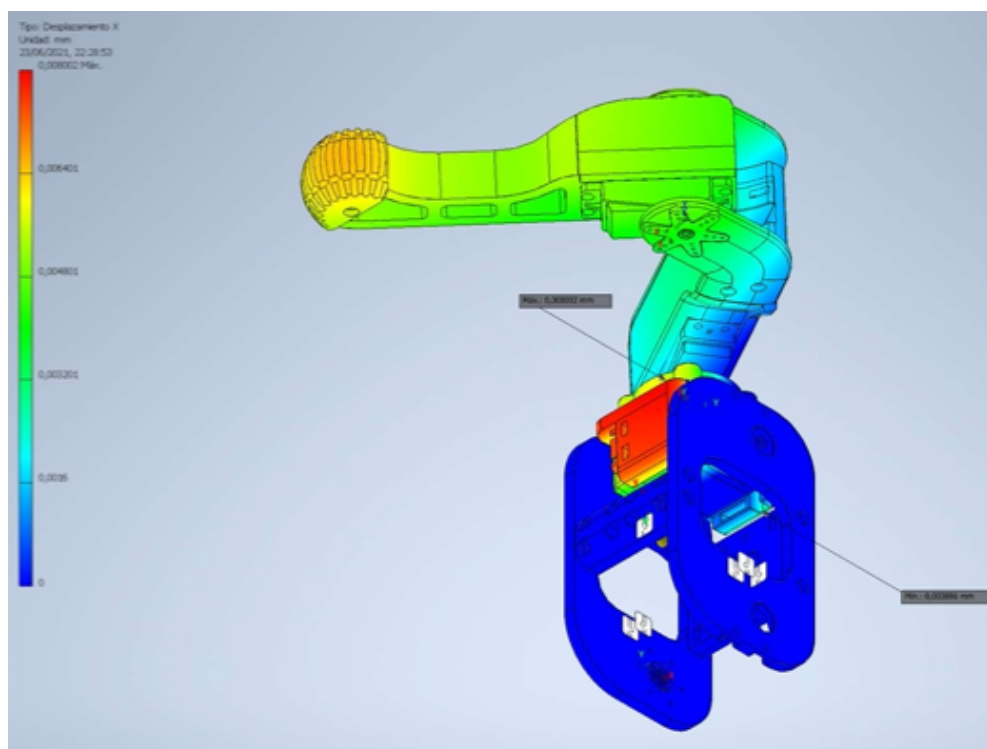


Figura 65. Ejemplo 1 de los resultados del análisis estático de tensiones distribuidas (Fuente propia).

Tras examinar los resultados de estas simulaciones, se obtiene que la tensión máxima sobre cualquier eje es de 25.72 Mpa con una tensión de Von Mises de 30.80 Mpa y siendo la mayor presión soportada de 85.70 Mpa. Teniendo en cuenta que la tensión principal máxima con la carga distribuida es de 25.72 Mpa, se puede verificar que las piezas estructurales no partirán bajo el mayor estrés, ya que el PLA tienen una resistencia mecánica a la flexión de 100 Mpa y a la tracción de 27-41 Mpa [59].

Por el contrario, el ABS tiene unas resistencias mecánicas de: 31- 86.2 Mpa para la compresión, 47.8-76 Mpa para la flexión y de 27.6-55.2 Mpa (Inventor emplea 20Mpa) para la tracción. Esto explica por qué al aplicar sobre una extremidad en la unión hombro abductor el peso completo de la articulación más los 15 kg·cm del propio servomotor, se superó el límite plástico y partió perpendicular al eje. Es decir, en la dirección en la que se aplicaba el esfuerzo. La solución para este componente (el *servohorn*), es emplear una versión de aluminio con una resistencia de un orden de magnitud superior. En lo referente a la fuerza, no se han superado los valores vistos en el análisis dinámico. Por tanto, se mantendrán los valores calculados en el apartado anterior.

#### 5.3.4 Elección de los nuevos componentes electrónicos y sensores

Una vez obtenidos los datos de los estudios anteriores, se puede proceder a la selección de los distintos componentes que sustituyan al resto de componentes.

En primer lugar, la **pantalla**. Originalmente se disponía de una *TFT* de 1.8", que será sustituida por una de 0.96" *OLED* con un protocolo de comunicación *I2C*. Esta última tiene un consumo de 3.3 V a diferencia de los 5 V anteriores. Esta tiene una resolución menor monocroma (128x64 píxeles), que no permite mostrar imágenes a color ni al tamaño de la anterior, pero permite la transmisión de información como el estado de la batería, valores del puerto serie, tiempo y programa en ejecución, entre otras. Por lo tanto, se ha mantenido el valor principal de la pantalla eliminando el uso de 6 puertos *GPIO*. Por último, se seguirá poder acceder a la visión de la cámara a través del servidor web.

En segundo lugar, será necesaria la adquisición de dos **sensores de ultrasonidos** HC-SR04 o superior y de un servomotor SG90 para desarrollar el sensor de distancia y detección por ultrasonidos (*USDAR*). Estos sensores servirán para la detección en un rango de 360° (180° por cada sensor) gracias al giro de 180° proporcionado por el servomotor de su base. Esto será explicado en detalle en el siguiente apartado. Este sistema también se valdrá de un anillo *LED* de 12 puntos *RGBW*, que permitirá proporcionar información visual de la detección o de otros estados de manera directa y a distancia independientemente del tamaño del texto o de la pantalla anterior.

En tercer lugar, está la **actuación principal**. Tras evaluar los resultados obtenidos en los distintos análisis se concluyó que sería necesario emplear un servomotor con unas especificaciones superiores a los previos. El modelo seleccionado es el DS3218MG de 180° y con un par en parada para 5V de 19 kg·cm y de 21.5 kg·cm con 6.8 V. El peso de estos motores es de 60 g, teniendo los engranajes de aluminio y acero, más resistentes y robustos ante la carga, problema que se encontraba con los anteriores de plástico y cobre. Por tanto, para el valor de tensión nominal de 6.5 V se obtiene el par suficiente para satisfacer los requisitos del prototipo, con un factor de seguridad de 1.55.

En cuarto lugar, se tendrán que realizar cambios en las **fuentes y sensores de alimentación**. Debido al cambio de servomotores, se ha decidido cambiar de la fuente principal de 9 A por una de 20 A, que acomode la nueva demanda de corriente requerida por los servomotores de mayor potencia.

Adicionalmente, se ha adquirido un sensor de corriente ACS712 de 30 A y un sensor de tensión de 25 V. Dichos sensores están sobredimensionados para el circuito, pero permitirán anomalías y obtener los mismos resultados que con un sensor a nivel, además de posibilitar el uso de fuentes externas para pruebas de larga duración y futuros cambios de la alimentación del circuito.

Por último, se deberá reemplazar el relé defectuoso y adquirir nuevos tornillos M2x8, ya que tras un uso quedan inservibles debido al tamaño de la boca. También se adquirirán bridas, y un sensor inercial que reemplace al anterior, del que se desprendió una resistencia debido a la fricción con el tornillo de montaje. Esto se subsanará empleando gomas anti-vibración para tornillos.

### 5.3.5 Diseño de las nuevas piezas estructurales

Tras los cambios realizados en los componentes electrónicos, fue necesario realizar algunos ajustes en los siguientes componentes estructurales:

- **Circuito v2:** el objetivo de este cambio consistió en ajustar las posiciones de los distintos componentes del circuito y sus tamaños (Figura 66). Se intercambiaron las posiciones de los transformadores con el fin de no desequilibrar el centro de masa en exceso, se modificaron sus puntos de sujeción, se ajustaron los puntos de sujeción del relé para el nuevo modelo, se subieron los puntos de anclaje de los terminales *USB* de acuerdo con los modelos empleados ya que antes eran poco accesibles y se recortó el sobrante destinado a luces *LED*. El circuito v2 se puede encontrar a continuación:

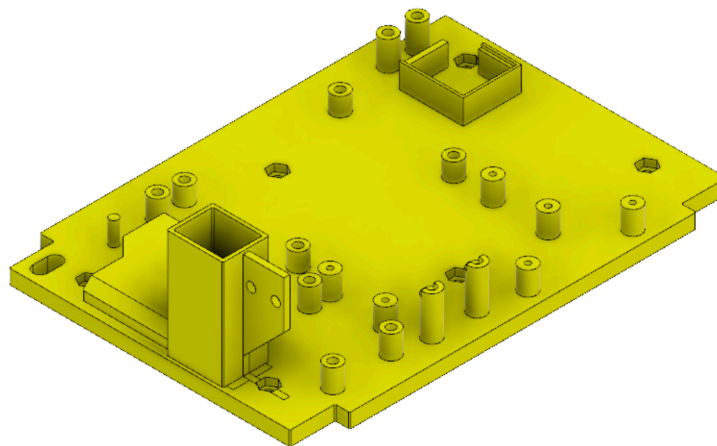


Figura 66. Circuito v2 (Fuente propia).

- **Cobertura trasera y montura de sensores:** en el modelo original estos componentes se encontraban unidos lo que es un inconveniente a la hora de reparar o modificar elementos. Por este motivo, se optó por un uso modular similar al frontal reduciendo el coste de plástico con los cambios. Para el soporte de sensores, se hizo el orificio para el botón, un soporte para el sensor de tensión (este cuenta con un agujero para tornillo central) y otro para el sensor de corriente, este encaja en hueco del soporte rectangular (Figura 67). El objetivo de las elevaciones es para tenerlos al mismo nivel que el botón una vez instalados (ver [vídeo de la construcción](#)). Los modelos finales se encuentran a continuación:

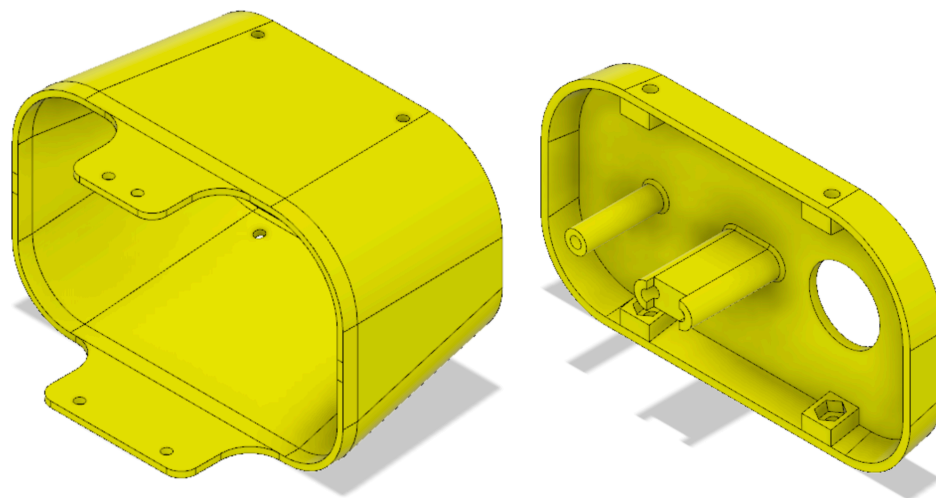


Figura 67. Cobertura trasera (izquierda) y montura para sensores de corriente, tensión y botón (derecha) (Fuente propia).

- **Tapa superior trasera:** por último, nos encontramos con la tapa trasera superior. La modificación en este componente es sencilla pues al desprenderse de la pantalla *TFT* de 1.8" e incorporar una *OLED* de 0.96", se necesitaba un punto nuevo de anclaje. Como el propósito era indicar el estado de la batería y el programa y en la parte trasera resultaría poco apropiado y supondría un problema con los sensores y el aumento de cables por el núcleo estrecho de los hombros, se decidió ponerlo junto con los puertos en una zona visible (Figura 68). Emplea 4 tornillos y tuercas M2 para su sujeción. El modelo se encuentra a continuación:

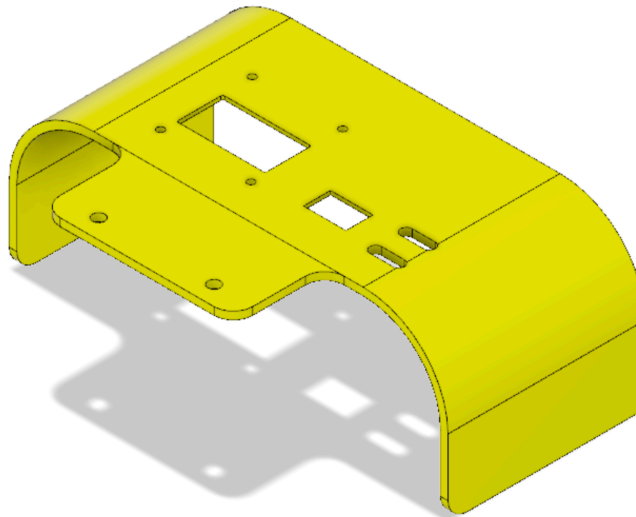


Figura 68. Tapa superior trasera v2 (Fuente propia).

### 5.3.6 Diseño del sistema de detección y medida de distancias mediante ultrasonidos (USDAR)

Tal y como se introducía en el apartado 5.3.1 de este capítulo, se propuso el diseño de un sistema similar a un *RADAR* o *LiDAR* convencional utilizando una serie de sensores de ultrasonidos sobre una plataforma móvil. El objetivo era tratar de cubrir los puntos ciegos presentes en el sistema de visión del robot que solo cubrían una vista de 95° de frente con los dos sensores *ToF* y la cámara. Para ello sería necesario desarrollar una torre montada en la parte superior para tener una visión completa y no bloqueada o colocar varios sensores en los laterales que podrían ser ocluidos. Se decidió apostar por el primero ya que proporcionaba el mayor rango de detecciones y permitiría “enfocar puntos de interés para calcular su distancia. Además de esto, garantizaba una zona de seguridad sobre el robot limitada a su distancia de detección efectiva, pero como se investigó en el capítulo tercero, podría ser ampliada con sensores más eficaces. El resultado debería ser muy inferior al coste de un *LiDAR* comercial. La imagen siguiente (Figura69) ilustra el planteamiento inicial de rangos:



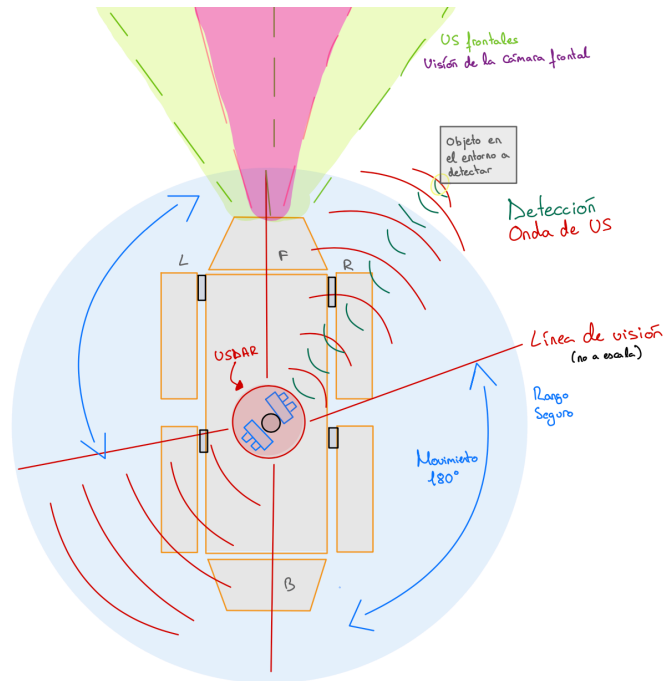


Figura 69. Primer boceto del USDAR y los rangos de visión del robot (Fuente propia).

Definido este, se comenzó a desarrollar una serie de prototipos y bocetos sobre la plataforma. Se llegó a la conclusión de que eran necesarias las siguientes partes:

- Una **base de apoyo sobre el robot** para acomodar el nuevo HW ya que el radio de giro con los sensores de ultrasonidos y un servomotor eran superiores a la zona plana disponible.
- Un **servomotor pequeño** que fuera capaz de mover el sistema.
- Un **sistema de soporte estructural del HW** a la base superior del robot para simplificar y mejorar las calidades de impresión. Esto se debe al hecho de que se pretende imprimir lo máximo posible sin soportes para minimizar posprocesado y errores de tamaño.
- Una **montura para los sensores de ultrasonidos**, este debería de anclarlos a un eje central sobre el que mantendría flotando para no aplicar rozamiento.
- Un **eje central** que conecte el servomotor (como un servohorn) con el resto de las componentes.
- Una **tapa superior** para evitar el acumulamiento de polvo y la entrada de agua directa a la zona de la electrónica, aunque su principal función es la estética y acoplar un anillo de LEDs que indique las detecciones.

Con estas ideas, se esbozó un primer diseño plano y otro tridimensional del concepto sobre el que basarse (figuras 70a y 70b) visibles a continuación:

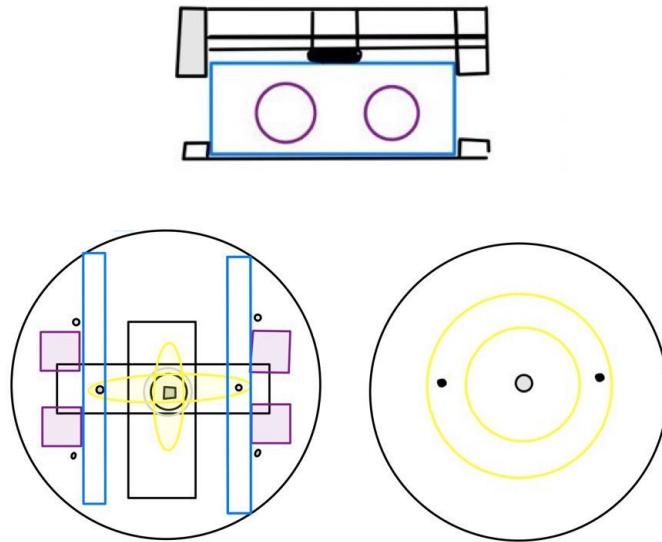


Figura 70a. Boceto del HW superior del USDAR. En la parte superior disponemos de la vista frontal. En la izquierda tenemos el motor en el centro con el eje de acople (negro) y el servohorn (amarillo) junto con los sensores de ultrasonidos. A la derecha se encuentra la vista superior con el anillo (amarillo) de leds anclado (Fuente propia).

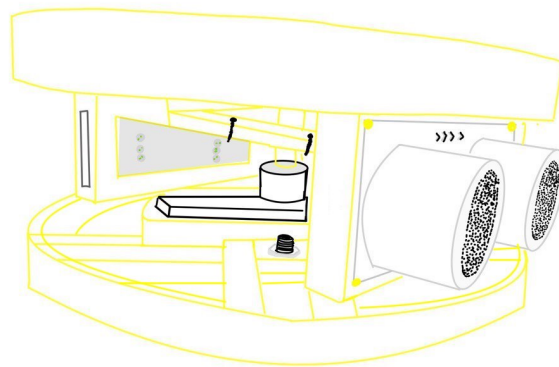


Figura 70b. Representación tridimensional de la Figura 70a (Fuente propia).

Una vez esbozado el concepto, se procedió al diseño de los componentes. El proceso seguido se detalla en mayor profundidad en el [vídeo de construcción](#) en el que se repasan las decisiones e iteraciones hasta llegar a las versiones actuales. Se detallarán a continuación las decisiones de diseño sobre los componentes finales haciendo referencia a las anteriores versiones sobre las que han sido iterados, así como a su funcionalidad conceptual.

En primer lugar, se diseñó el **soporte estructural del HW** (Figura 71). Como se ha mencionado, su objetivo consistía en servir como una zona de anclaje del resto de los componentes a la tapa superior. Para ello se realizaron una serie de mediciones sobre las alturas de los componentes, así como del radio de giro que ocuparía cada uno. Con ello se obtuvo la circunferencia máxima del modelo. Sobre esta, se extruyó un muro o borde con un eje central apoyado por nervios estructurales. Esto se hizo de esta forma para ahorrar material y por ende peso al robot, manteniendo una buena rigidez estructural. El muro además sirvió como cobertura visual para el hemisferio inferior del sensor y apoyo en caso de que la unión superior flexase para los sensores ToF. Seguidamente, en el centro, se crearon los puntos de anclaje a la estructura inferior, así como el hueco y anclajes que acomodaría al servomotor. En este caso, se trata de un SG-90 de 3kg\*cm con un par  $\sim 3$  veces superior al requerido. El servomotor destaca por su pequeño tamaño que lo hace perfecto para esta aplicación, ya que el espacio en la zona del circuito es de por sí limitado y un tamaño superior generaría una estructura superior más elevada de forma innecesaria. Se iteró con la altura y profundidad del servomotor hasta conseguir en conjunto con el anclaje de los sensores, una altura idónea en la realidad y en el modelo ensamblado. El resultado es el siguiente:

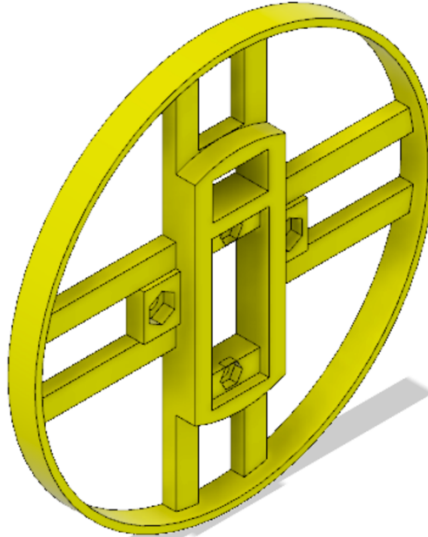


Figura 71. Modelo de la base de montaje (servoradarbase\_v14.step) (Fuente propia).

Conseguida la pieza de anclaje, el siguiente paso era su montura sobre el robot real en la **tapa superior frontal** (Figura 72). Para ello fue necesario modificar la superficie del lomo superior. Se realizó una extrusión circular del mismo diámetro y se recortaron los orificios correspondientes a los tornillos estructurales que atornillarían desde la parte inferior anclando con las tuercas

del elemento de soporte, otro para el cableado y el hueco del propio servomotor. El modelo no sufrió muchas iteraciones, aunque el orificio para el servomotor era muy apretado y de por sí sujetaba al sistema con firmeza. Se limó un poco para su correcta inserción y se modificó el archivo 3D para futuras impresiones. El resultado es el siguiente:

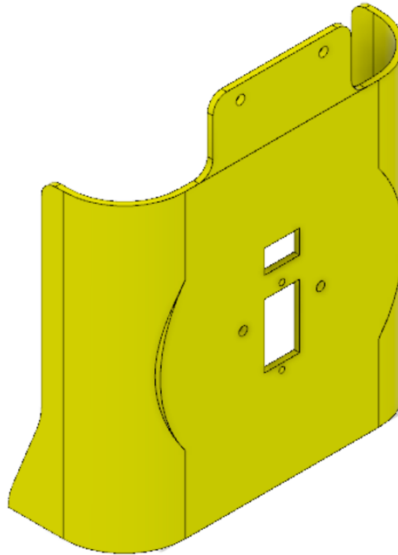


Figura 72. Modelo de la tapa superior frontal (top\_front\_cover\_v2.step) (Fuente propia).

El siguiente paso era la elaboración del **núcleo de soporte central** (Figura 73). Para este diseño se partió de un modelo previo de servohorn para SG-90 y se adaptó para soportar un elemento superior. Esta es la pieza con más cambios debido al hecho del calibrado de distancias de anclaje con las siguientes piezas. El diseño consiste en un elemento de servohorn con un orificio para un tornillo que sujete el eje al servomotor, dos brazos con orificios para tuercas M2 sobre los que se acoplan las sujeciones de los sensores de ultrasonidos, y un soporte superior para una tuerca y guía para la tapa superior. El modelo también cuenta con una tapa de plástico con forma de nuez M3 que fusionar sobre la tuerca y las paredes, permitiendo montar la parte superior del sistema al ejercer fuerza reactiva sobre la tuerca al atornillar. El modelo es el siguiente:

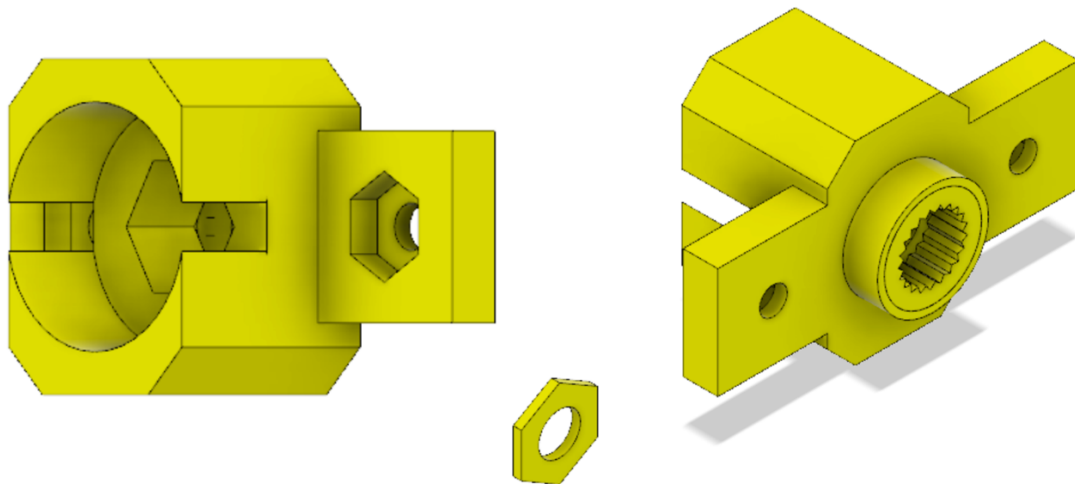


Figura 73. Modelo del núcleo de soporte (nucleo\_v9.step). A la izquierda la vista superior, en el centro la cubierta para la tuerca M3 y a la derecha el enganche del servohorn (Fuente propia).

El siguiente paso natural fue desarrollar el **soporte principal del sensor de ultrasonidos** (Figura 74). Para el diseño de este, se partió del modelo 3D disponible en GrabCAD y se generó una estructura tipo esqueleto de soporte con una cavidad central para los componentes electrónicos sobresalientes de la placa. Para ello, se apoyaría con 4 mini torres con orificios para tornillos M1 por los que se acoplaría el sensor. Sobre esto, crecería un nervio extruido justo al borde de los pines Dupont que conectaría con la pieza anterior. Esto es posible gracias a un orificio M2 y una cavidad de apoyo (hembra) con el tamaño del ancho del núcleo estructural (macho). Las iteraciones consistieron en obtener una relación de anchura y longitud precisa. Hay que destacar que es la única pieza que necesita soportes, aunque estos son mínimos y no afectan el producto. La pieza final es la siguiente:

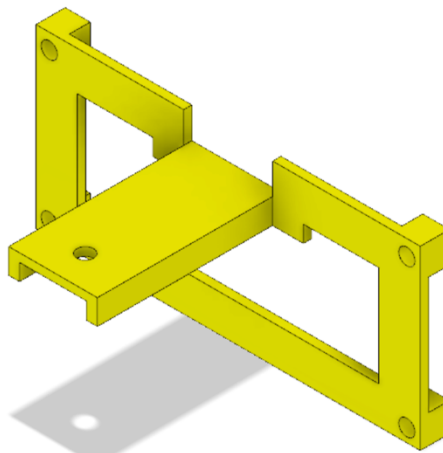


Figura 74. Modelo de la pieza de soporte de los sensores de ultrasonidos (us\_support\_v6.step) (Fuente propia).

Es en este momento en el que surge la solución a la falta de tornillos M1 motivada por una mejora estructural. La idea era diseñar una **tapa frontal** (Figura 75) para el sensor de ultrasonidos que cubriese el sensor dejando únicamente las soldaduras y los componentes protuberantes visibles. La pieza aprovecharía su precisión para engancharse a la pieza madre e incluiría unos agarres laterales. De nuevo y siguiendo el problema de los tornillos **M1**, se decidió utilizar un **perno** de plástico que se soldaría/fusionaría con un **remache** trasero para asegurar el sensor y sus soportes. Inicialmente se creó un tornillo M1 y una tuerca M1 con CAD, pero debido a la falta de boquillas de diámetro 0.2, no fue posible alcanzar la precisión requerida. Por lo que se decidió emplear lo último. Las imágenes de los modelos en CAD se encuentran a continuación:

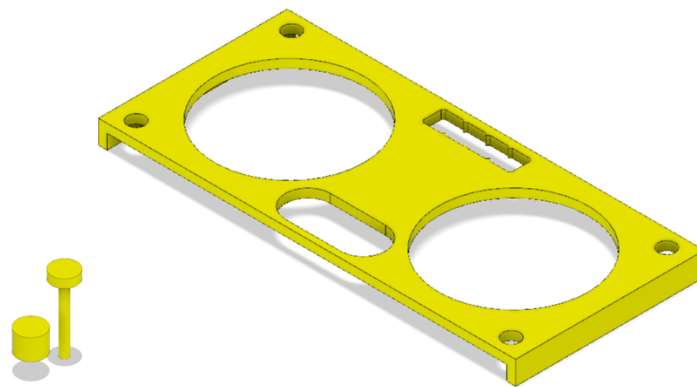


Figura 75. A la izquierda el modelo del perno y remache (pernos\_v6.step) y a la derecha el modelo de la tapa frontal del sensor de (*us\_front\_v4.step*) (Fuente propia).

Finalmente tenemos **la tapa superior del sistema** (Figura 76) y el componente que más modificaciones ha sufrido. Esto se debe a que originalmente estaba compuesto por dos partes ya que el nervio central era hueco y utilizaba un eje y un rodamiento superior para anclarse a la base. No obstante, debido a fallos y a la poca utilidad, se modificó por el sistema actual más sencillo y de una única pieza. La tapa actual cuenta con un orificio para un tornillo M3 que queda completamente escondido, un anillo de soporte a presión para el anillo de luces LED con su tapa siendo de 0.2 mm de espesor (la altura de capa) para permitir que la luz traspase claramente. También sigue la dinámica de la base con un anillo exterior que camufla el conjunto y cuenta con un nervio central que encaja en el orificio del núcleo de soporte y aloja una tuerca adicional. A continuación, se puede observar una ilustración de la última versión (nº 19):

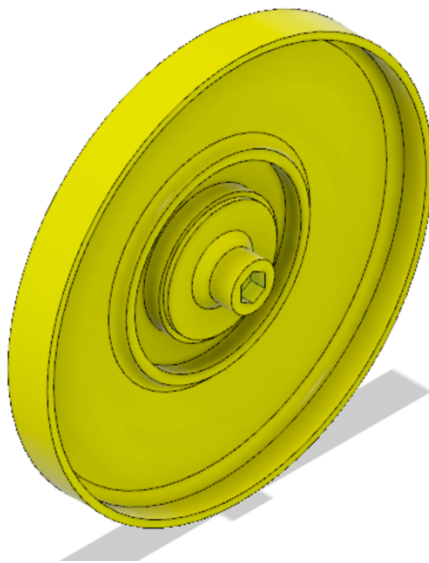


Figura 76. Modelo de la tapa superior (*topradar\_v19.step*) (Fuente propia).

### 5.3.7 Ensamblado y comprobación del modelo virtual

Una vez diseñadas las piezas, es necesario ensamblarlas para verificar su operabilidad. En este caso, tras haberse modificado componentes de sujeción electrónica, no es necesario ensamblarlas al modelo, pues se han modelado empleando modelos virtuales de los mismos sin alterar los tamaños.

Sin embargo, en el caso del sistema *USDAR*, al ser móvil, sí es necesario verificar que la suma de las partes funciona de acuerdo con las restricciones cinemáticas del modelo. Para ello se procedió a su ensamblaje y verificación de operabilidad. En esas pruebas, se comprobó cómo podría existir un problema estructural en el diseño, ya que el punto de apoyo y el sistema de enganche original (primera implementación), resultaron poco prácticos para el acoplamiento al servomotor de rotación. En estos ajustes se simplificó el modelo reduciendo el número de componentes simplificando así el sistema tal y como se ha mencionado en el apartado anterior.

Una vez hecho esto, el sistema parecía más estable y se redujeron los tiempos de impresión. No obstante, sería necesario realizar unas últimas pruebas de impresión para verificar los acoplamientos. El modelo ensamblado se puede observar en las siguientes figuras (77 y 78) (el componente “*topradar*” mostrado corresponde a la versión 15 no 19):

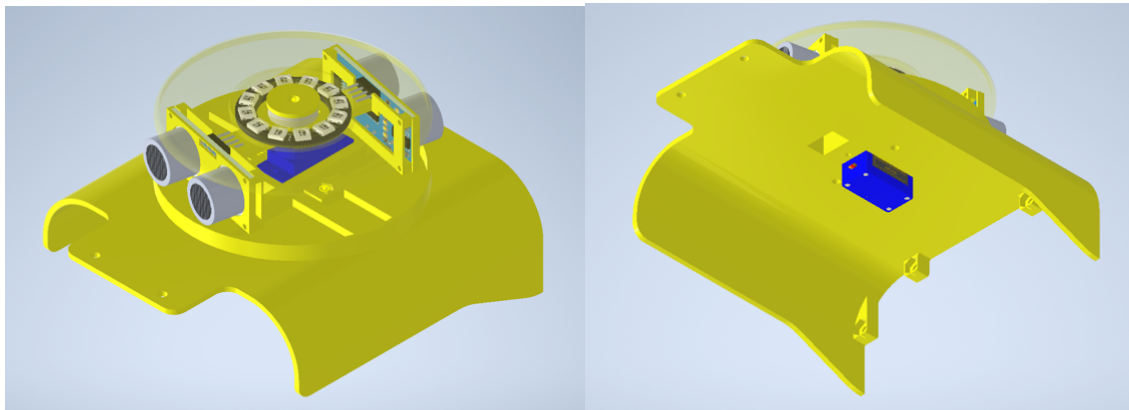


Figura 77. Modelo del USDAR en su versión 2ª ensamblado en vista superior e inferior (Fuente propia).

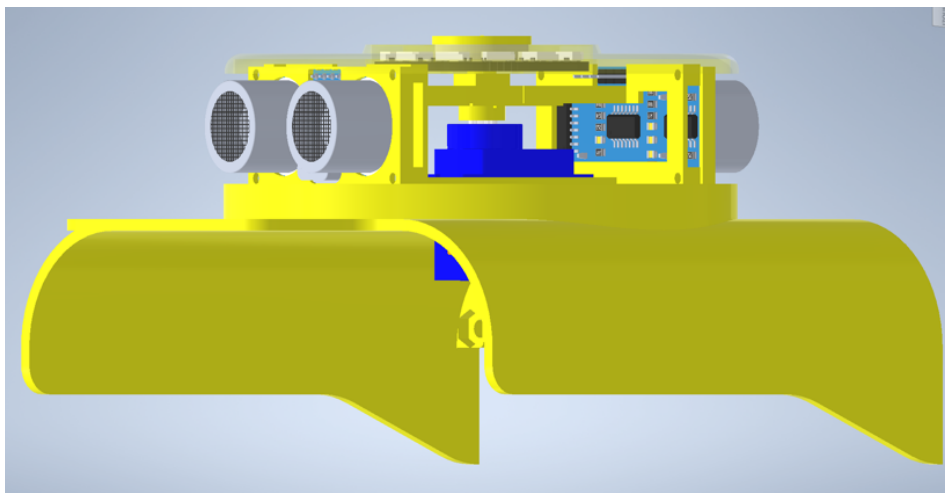


Figura 78. Modelo del USDAR en su versión 2ª ensamblado en vista lateral (Fuente propia).

### 5.3.8 Revisión del esquema electrónico

Teniendo en cuenta la modificación realizada en la lista de los componentes empleados, fue necesario actualizar el circuito para acomodar en un primer lugar los nuevos componentes del sistema base, y en un segundo lugar el sistema *USDAR*, así como reestructurar el diagrama de conexiones haciendo que todos los cables salvo los servomotores se encuentren completamente dibujados. De esta forma se pueden visualizar aquellos cables que por conveniencia se han combinado en una única salida, aún teniendo distintos puntos de entradas correspondiente a distintos componentes electrónicos. También son visibles los nuevos componentes correspondientes a los 2 sensores de ultrasonidos del *USDAR* y el servomotor SG-90.



A continuación, se encuentra la leyenda de colores para los cables y las distintas versiones del circuito modificadas para esta segunda fase (figuras 79a y 79b):

- Negro y marrón: tierra o GND.
- Cian: polo positivo de la batería 8.4V.
- Azul oscuro: polo positivo de 6.5V.
- Violeta: polo positivo de 5V.
- Rojo: polo positivo de 3.3V.
- Blanco: I2C SDA.
- Gris: I2C SCL.
- Verde: Echo US
- Amarillo: Trigo US
- Otros: cables empleados para señales digitales.

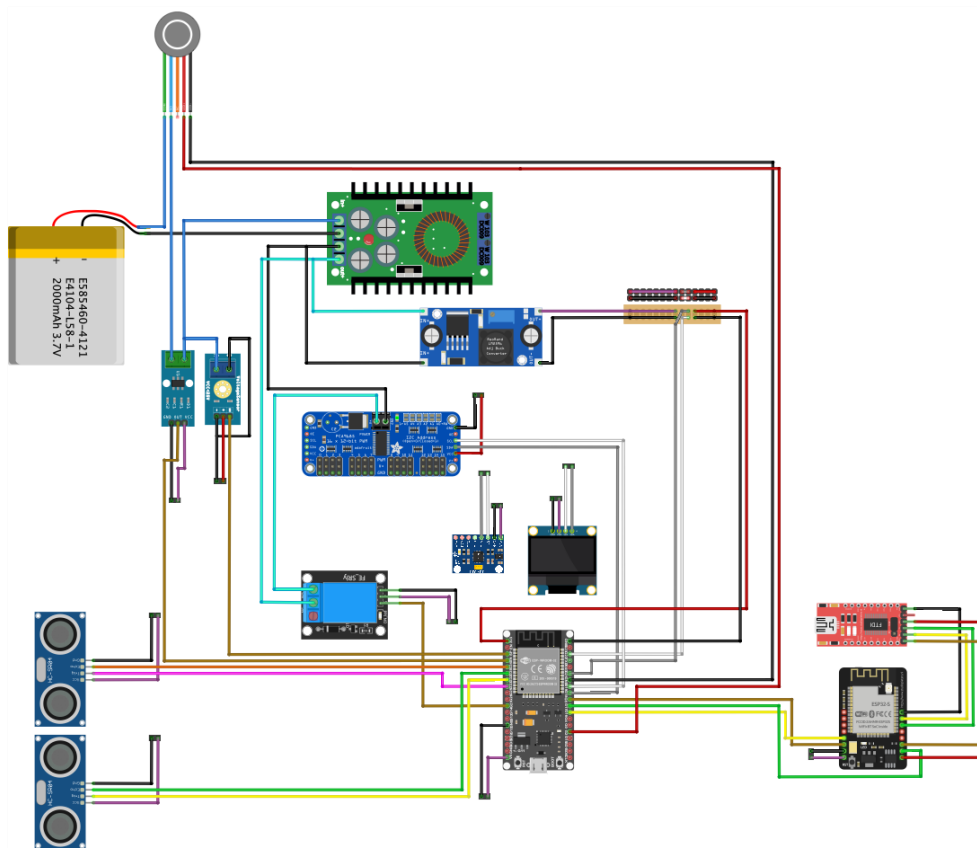


Figura 79a. Diagrama de circuito v2 (Fuente propia).

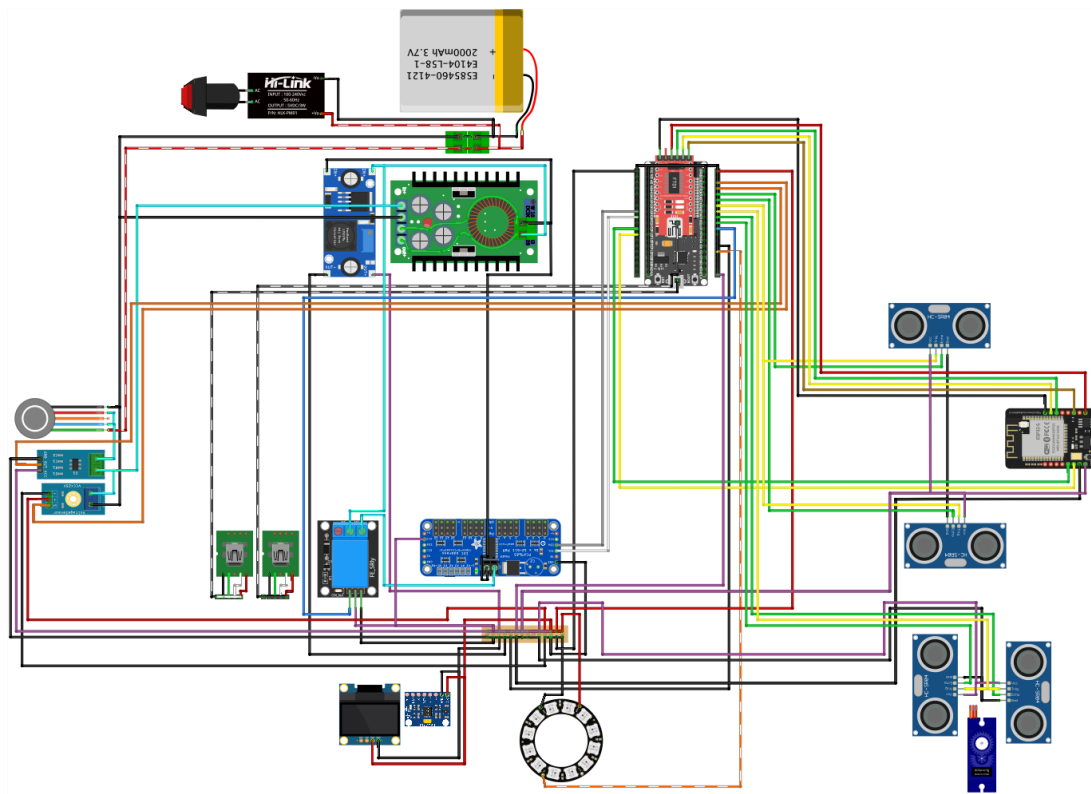


Figura 79b. Diagrama de circuito v3 (Fuente propia).

### 5.3.9 Revisión de las tablas de costes e impresión

A continuación, tras recopilar la información de los cambios en los componentes necesarios y de las nuevas impresiones, se obtienen los costes totales. Se han marcado en rojo aquellos elementos que han sido devueltos o sustraídos del total final de costes. No se han tenido en cuenta las impresiones de prueba de los componentes, estos estarán disponibles en la hoja final correspondiente a los costes totales del proyecto. Las tablas 6 y 7 con los nuevos costes de material e impresión, se encuentra a continuación:

Nombre	Modelo	Cantidad	Precio €	Total €
Servomotores	MG996R Servo 12Kg	11	-5	-55
Servomotores	DS3218MG Servo 20kg	12	11	132
Pantalla	0.96" OLED I2C Display	1	10	10
Sensores de distancia	HC-SR04 Ultrasonic Sensor	2	3	6
Sensor inercial	GY-521 Gyroscope and Accelerometer	1	4	4
Relé	HW-482 / KY-019 5V 10A Relais	1	8	8
Transformador principal	XL4016 DC-DC 9A Stepdown Converter	1	-14	-14

Transformador principal	SZBK07 DC-DC 20A Stepdown Converter	1	16	16
Sensor de corriente	ACS712 30A current sensor module	1	6	6
Sensor de tensión	25V 125voltaje sensor module	1	6	6
Pantalla	1,8" TFT with ST7735 w/o SD	1	-14	-14
Bridas	Bridas de 20cm	1	5	5
Tornillos	M2x8 cylinderhead screws + M2 nuts (100)	1	8	8
Total	No incluye el gasto de luz			118

Tabla 6. Costes de los materiales para la segunda fase (Fuente propia).

ALFREDog – FASE 2 – COSTES DE IMPRESIÓN													
USDAR													
Cantidad	Nombre de la pieza		Relleno	Duración	Filamento		Total de la duración	Total de filamento					
2X	Soporte frontal de US		20%	1h 56min	15 g		4h	30 g					
8x	Pines de soporte		100%	10min	1 g		20min	2 g					
1x	Soporte central		20%	48min	9 g		1h 32min	9 g					
2x	Soportes laterales de los US		100%	2h 42min	27 g		5h 24min	54 g					
							Total	95 g					
Cubiertas del chasis y circuito													
Cantidad	Nombre de la pieza		Relleno	Duración	Filamento		Total de la duración	Total de filamento					
1x	Cubierta frontal		100%	6h 2min	63 g		6h 2min	63 g					
1x	Cubierta trasera		100%	4h 37min	53 g		4h 37min	53 g					
1x	tapa superior frontal		100%	5h 20min	71 g		5h 20min	71 g					
1x	tapa superior trasera		100%	2h 21min	28 g		2h 21min	28 g					
1x	Soporte superior		100%	4h15min	46 g		4h15min	46 g					
1x	Soporte pines de alimentación		100%	3h 7min	28 g		3h 7min	28 g					
1x	Circuito v2		20%	8h 14min	47 g		8h 14min	47 g					
							Total	336 g					
							<table><tr><td>Total PLA Amarillo</td><td>431</td></tr><tr><td>Total de filamento</td><td>431</td></tr></table>			Total PLA Amarillo	431	Total de filamento	431
Total PLA Amarillo	431												
Total de filamento	431												

Tabla 7. Costes temporales y materiales de impresión para la segunda fase (Fuente propia).

Tras esto, se procedió a la impresión de las distintas piezas y, en los casos del USDAR, fueron necesarias varias impresiones hasta conseguir una versión correcta y funcional. Esta capacidad para poder iterar y probar piezas solo es posible gracias al uso de las tecnologías de impresión 3D y permite solucionar

aquellos problemas que no son visibles en el modelo 3D, como en el caso de la unión del núcleo y la tapa del USDAR.

## 5.4 Construcción y verificación: Fase 2

### 5.4.1 Cambios en el cableado, componentes y distribución interna

Tras el cambio completo del circuito, fue necesario rehacer el 90% del cableado previo para acomodar las nuevas distancias y sensores, en este caso, de tensión y corriente. El nuevo cableado consistía en una cadena de conexiones entre los sensores con una tierra común (Figura 80). En el centro del circuito, con la nueva distribución, fue necesario enrutar las líneas del polo positivo y negativo por dos zonas distintas tal y como se puede apreciar en la Figura 81. Se tuvieron que cambiar los terminales de conexión del módulo SZBK07 ya que el tipo de conexión era propenso a cortocircuitar (dañando el *step down* menor). Los nodos se sacaron de la vieja placa de control de la impresora 3D que fue sustituida por una de 32 Bits, de nuevo, las imágenes de todo el proceso así como el vídeo de la construcción, están disponibles en la [carpeta en la nube](#).

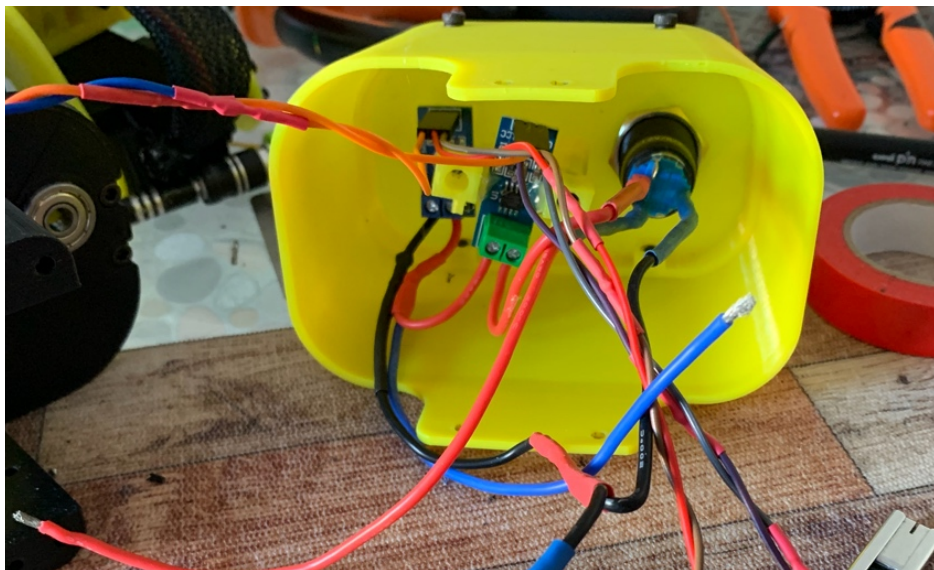


Figura 80. Circuito interruptor con sensores v2 (Fuente propia).

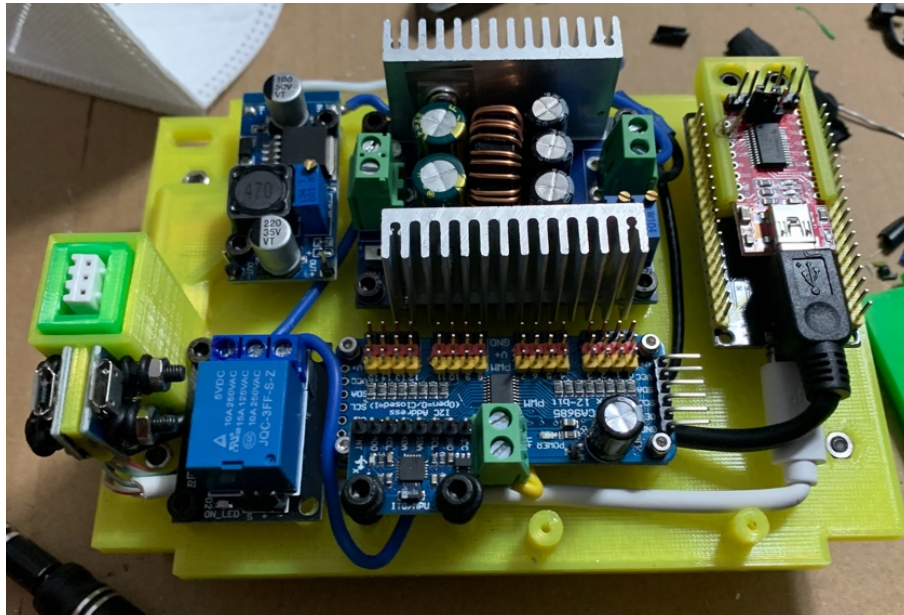


Figura 81. Circuito principal ensamblado v2 (Fuente propia).

#### 5.4.2 Elaboración de códigos de testeo

Al realizar cambios y añadir nuevos componentes, fue necesario elaborar nuevos códigos de prueba. En este caso, se elaboró un código para verificar cada uno de los nuevos componentes:

- **Sensor de tensión:** el código consistía en una simple regla matemática para extraer la tensión a partir del divisor de tensión con las 2 resistencias incorporadas.

$$\text{Tensión de entrada} = (\text{valor} * 5.0) / 1024.0;$$

$$\text{Tensión de salida} = \text{Tensión de entrada} / (R2/(R1+R2));$$

No obstante, en la fase 3, se modificaría para extraer la corriente estimada empleando un código más complejo basado en la capacidad para procesar la intensidad.

- **Sensor de corriente:** el código, de nuevo, seguía la fórmula matemática proporcionada por el fabricante:

$$\text{Valor de corriente} = (2.5 - (\text{Media de valores} * (5.0 / 1024.0))) / 0.066;$$

- **USDAR:** para controlar el sistema de detección, se programó un ciclo de rotación constante de 180° en el que se realizaba una medida por cada

grado de giro. El resultado de la ejecución es visible en el [vídeo en la carpeta compartida](#).

- **Anillo de LEDs:** con el fin de verificar el funcionamiento del anillo, de nuevo, empleamos la librería del fabricante para gestionar cada uno de los doce LEDs de manera independiente.

Todos los códigos arriba mencionados, se encuentran referenciados en el [Anexo 1](#).

#### 5.4.4 Pruebas de funcionamiento de sensores y electrónica

Con el circuito montado y los programas listos, se procedió de nuevo a realizar las pruebas de verificación de los nuevos componentes antes de el ensamblaje final. El objetivo es detectar componentes defectuosos y comprobar que dichos componentes presentan un comportamiento esperado. Para dichas pruebas se usó una placa Arduino uno pues el cableado de los sensores no se había conformado.

En primer lugar, se realizaron pruebas sobre los sensores de tensión y corriente, los resultados fueron satisfactorios y se pudieron medir las tensiones de forma correcta. Estas mediciones se realizaron sobre una fuente de alimentación casera regulable de 3 a 6.6 V y posteriormente empleando la propia batería interna del robot. En lo referente al relé sustituido, de nuevo bastaba con un simple comando de bit alto o bajo. Tras esta prueba, se comprobó que el relé funcionaba correctamente y por tanto el problema anterior quedó resuelto. Se calibraron los transformadores internos y se verificaron las tensiones como se puede ver en la Figura 82.

Tras realizar esto, se comprobó el anillo de luces LED, el cual resultó tener un defecto en el LED rojo de una unidad LED. Por lo demás, funcionaba correctamente, aunque sería reemplazado por otro. En lo referente a la nueva pantalla, se empleó el código del fabricante y resultó no funcionar. El problema detectado fue que, aunque en las especificaciones esta funcione con una tensión de entrada de 3.3 a 5 V, en la realidad solo funcionaba con 3.3 V.



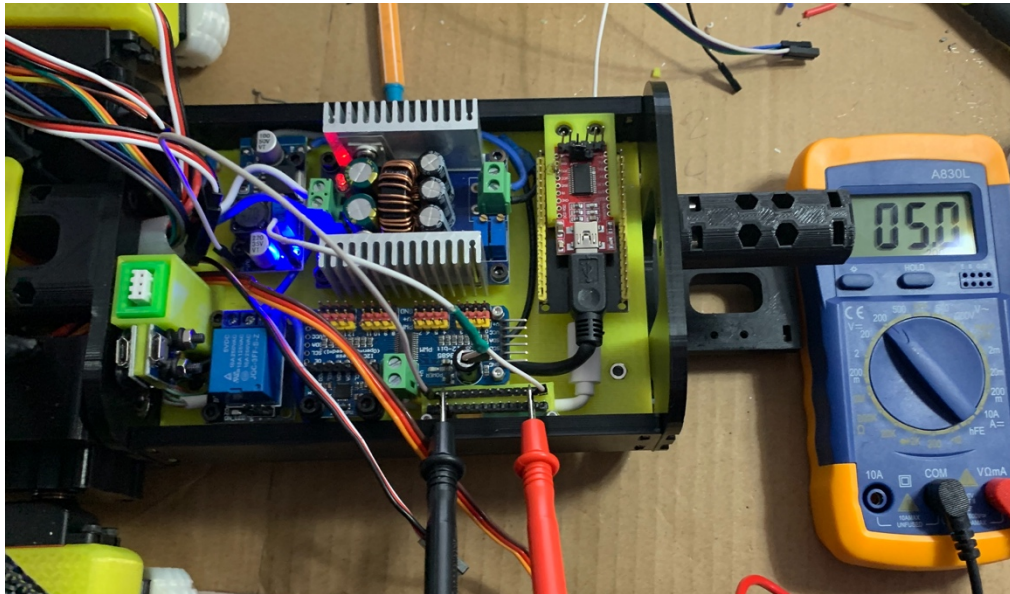


Figura 82. Verificación de tensiones en el circuito principal v2 (Fuente propia).

Una vez verificado el circuito principal, se comprobó que todos los servomotores alcanzasen, al mismo tiempo, la posición máxima comandada empleando el mismo código que en la fase anterior. Tal y como era de esperar, había pequeñas diferencias en el movimiento debido a los pequeños errores en la fabricación. Por ello se distribuyeron en 3 grupos de 4 servomotores (Figura 83) que serían instalados en los distintos grados de libertad del robot. De esta forma mitigamos esta discrepancia al homogeneizarla en las cuatro cadenas cinemáticas del robot. Adicionalmente, se verificó también el funcionamiento del pequeño servomotor, el SG-90, que iría ubicado en la parte superior con el USDAR, verificando su código de movimiento.



Figura 83. A la izquierda la placa Arduino empleada para la verificación de los servomotores y agrupación en grupos (centro) con las patas de la fase 1 a la derecha. (Fuente propia).

### 5.4.3 Montaje del robot

Una vez comprobados todos los sistemas, llegó el momento del montaje del prototipo de la fase 2. Para lograrlo fue necesario desensamblar todo el robot salvo por la cabeza que no tuvo cambios. Este proceso, como el anterior, es descrito en el [vídeo de explicaciones de la construcción](#). Una vez desmontado, se sustituyeron los servomotores y se reemplazó la parte trasera del robot. Fue aquí cuando se comenzó a iterar sobre el diseño del *USDAR* detectando una serie de fallos en el nervio central. Resultó que el hecho de ser modular generaba que el punto de agarre no fuese lo suficientemente estable y no se ejerciera un agarre por presión de la tapa al quedar semisuspendida (Figura 84 ) como se puede ver a continuación:

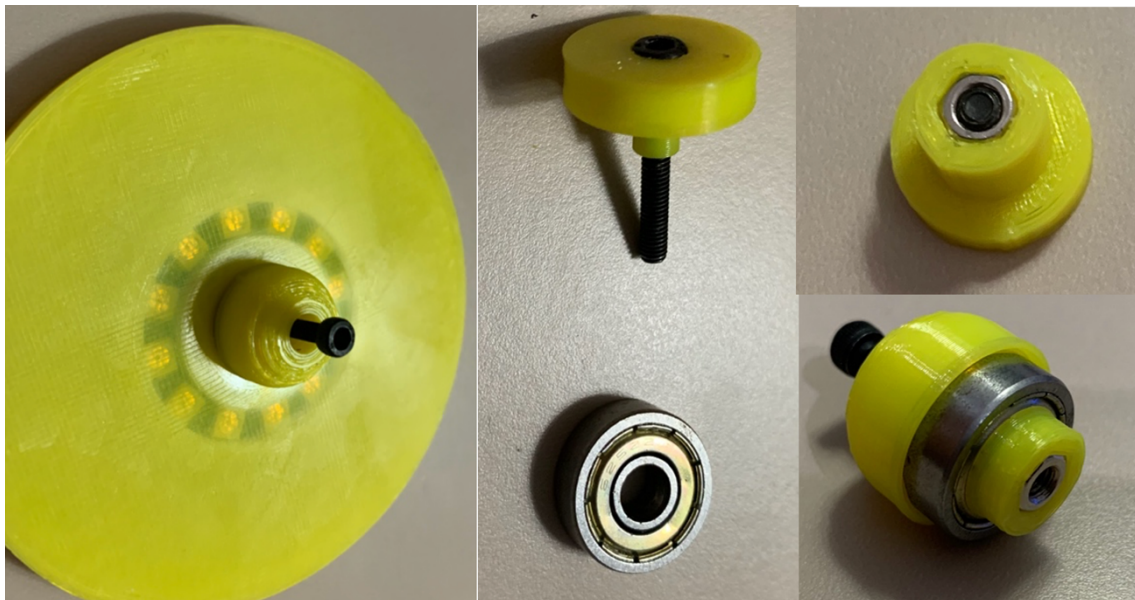


Figura 84. Compuesto de distintas pruebas de soporte de la tapa superior con el nervio central (Fuente propia).

Tras 4 iteraciones principales y algunos ajustes en el diseño, se obtuvo la versión actual mencionada en el punto 5.3.6. Con estos cambios se obtuvo una versión mucho más simplificada del mismo (Figura 85-87) de la que podemos apreciar sus distintos componentes en distintas fases de implementación a continuación:



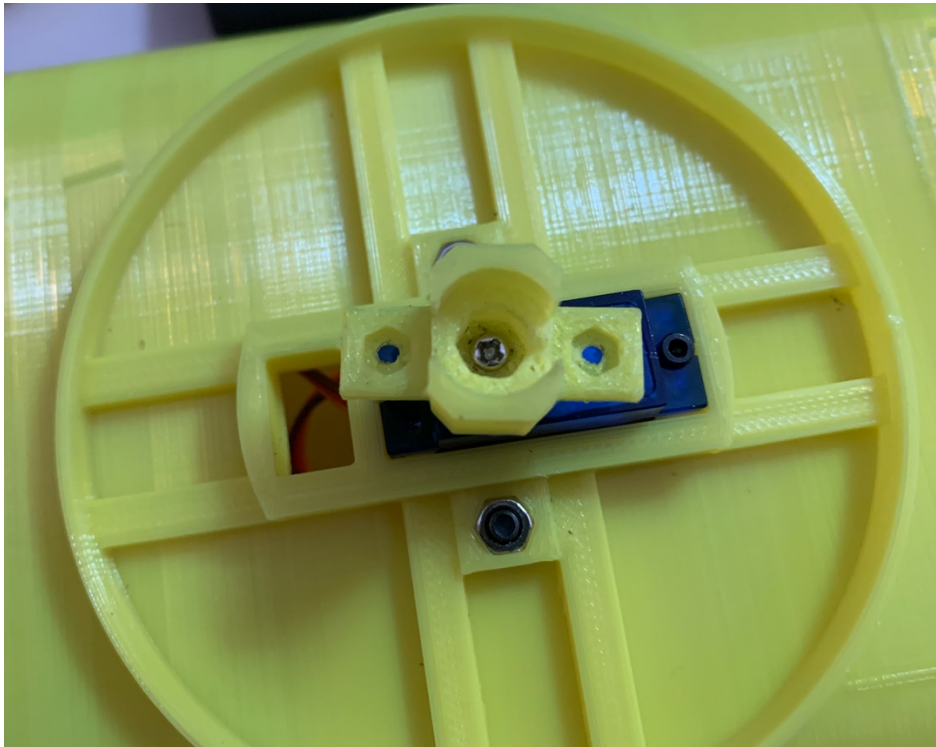


Figura 85a. Instalación del núcleo principal (Fuente propia).

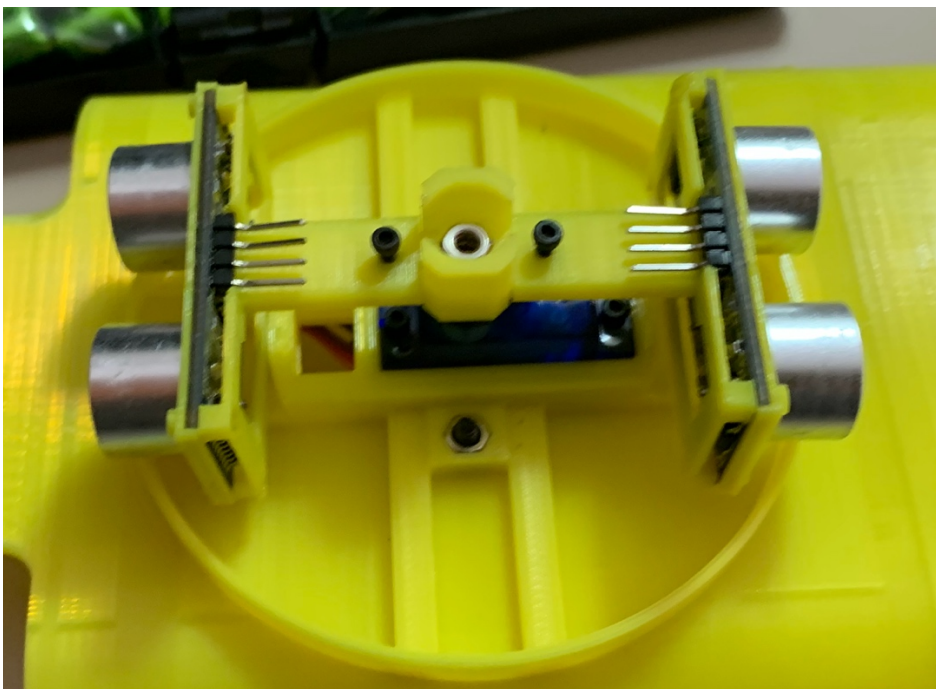


Figura 85b. Instalación del núcleo principal y el soporte de los sensores de ultra sonidos(Fuente propia).

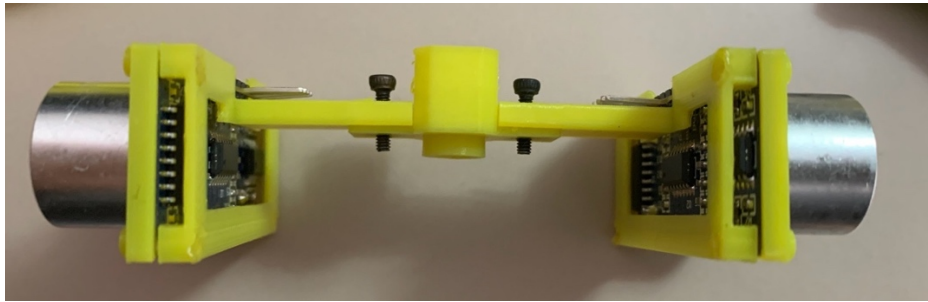


Figura 86. Núcleo principal junto con el soporte de ultrasonidos en vista lateral (Fuente propia).



Figura 87. Instalación del núcleo principal y la tapa superior con el anillo de LEDs (comprobación de estabilidad) (Fuente propia).

Adicionalmente, se imprimió y construyó una versión sin el soporte para el *USDAR* que serviría para pruebas con el fin de evitar dañarlo (Figura 88) integrándolo también en el robot como se puede ver a continuación:

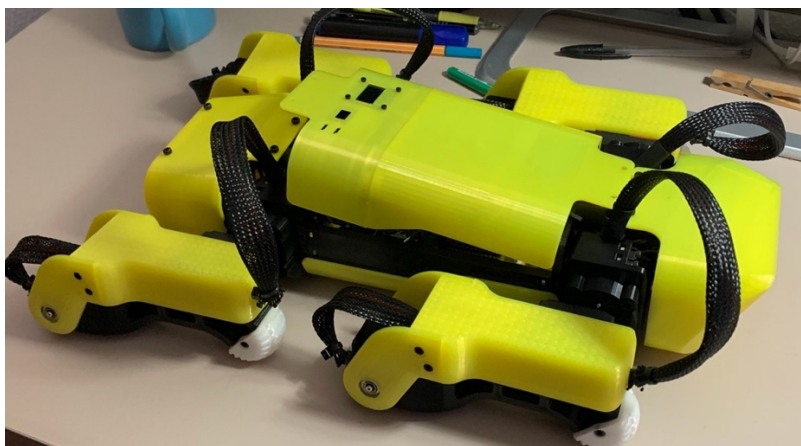


Figura 88. Presentación sobre el robot F2 de la tapa sin *USDAR* (Fuente propia).

Finalmente, se implementó también una carcasa superior para alojar el *USDAR* con el fin de ser intercambiable en cualquier momento (Figura 89) como se puede apreciar a continuación:



Figura 89. Presentación de la versión *USDAR* sobre el robot de la fase 2 (Fuente propia).

#### 5.4.5 Evaluación del movimiento de las articulaciones, *USDAR* y componentes

Con el robot ensamblado, se verificó que el cambio de los servomotores producía movimientos continuos y con fuerza suficiente como para aplicarle peso. De nuevo, se probó con un código de posicionamiento y con el de control por joystick de los ángulos. El *USDAR* por otro lado, se probó de forma individual y se pudo verificar que efectivamente, era capaz de cubrir todo el rango de movimientos como se puede ver en uno de los vídeos de prueba referenciados en el [Anexo 1](#) y en la Figura 90:

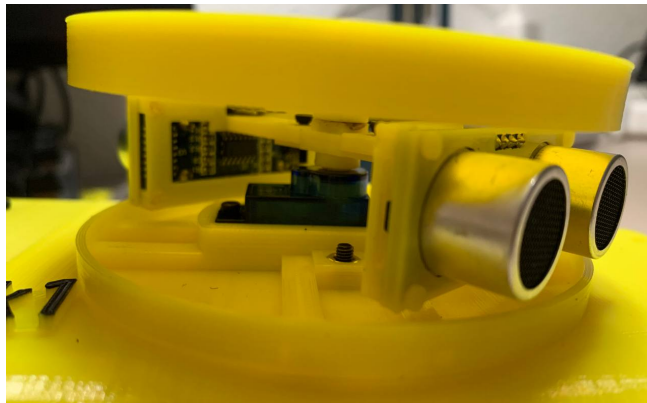


Figura 90. Pruebas del USDAR sobre el robot de la fase 2 (Fuente propia).

## 5.5 Diseño e impresión del prototipo: Fase 3

Una vez conseguida la base deseada en la fase 2, llegó el momento de comenzar la implementación de la fase 3. En esta fase se pretende como se mencionaba al comienzo del capítulo, implementar modificaciones e incorporar nuevos sensores que permitan en un futuro lograr autonomía y un control inteligente.

### 5.5.1 Planteamiento de mejoras al sistema

El primer paso consistió en evaluar las carencias y posibles mejoras a incorporar al sistema en base a la plataforma actual. Esta nos permite mover al robot y evitar choques, pero solo tenemos información útil del *IMU* y los sensores de ultrasonidos. No hay forma de, como con los sistemas comerciales, conocer el posicionamiento exacto de las articulaciones al carecer de los encoders. Sin este conocimiento, el control avanzado sería imposible ya que no podemos verificar el éxito o compensar un imprevisto en el posicionamiento. Por ello era necesario idear un sistema alternativo. El objetivo de este sistema era emular la información de un encoder, para tratar de conocer la posición espacial de las articulaciones con respecto al sistema de referencia del centro del robot.

El sistema alternativo requería fusionar los conocimientos de la cinemática inversa con métodos de trigonometría vectorial auxiliares. La premisa era sencilla, conociendo la orientación del antebrazo, un punto de contacto y un punto fijo de referencia, se podría conocer la cadena cinemática completa (Figura 91) como se puede ver a continuación:

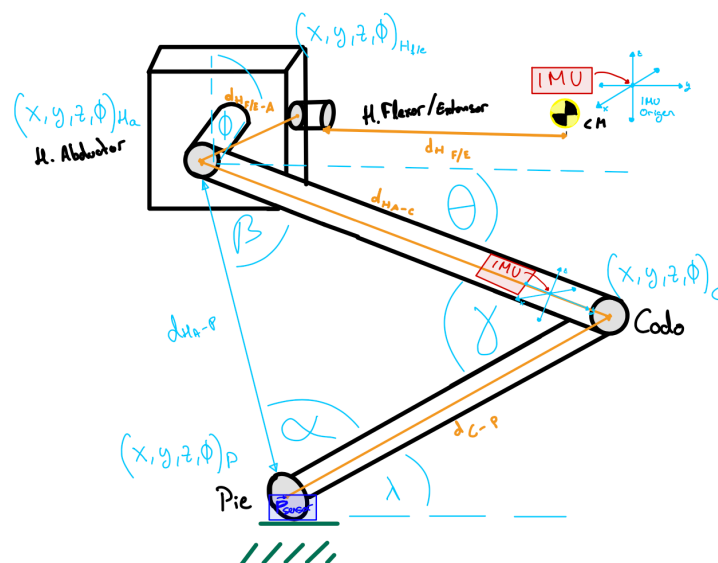


Figura 91. Esquema del sistema IMU-ENCODER (Fuente propia).



La solución planteada (Figura 93) consistía en incorporar sensores de presión en los puntos de apoyo de las patas y unos sensores inerciales en los brazos (entre el hombro y el codo). Con estos sensores podríamos conocer mediante los ángulos posibles de la cinemática (Figura 91) y ecuaciones trigonométricas 2 posibles soluciones para cada una de las cadenas. En la primera el robot se apoya como en la Figura 91 y en la segunda el antebrazo estaría invertido horizontalmente sobre el suelo como se puede apreciar en el resalte en morado de Figura 92. Sin embargo, debido a que de ser este el caso, el robot no podría mantenerse estable, aunque el punto de presión pudiese llegar a activarse, sería descartado por el controlador. Por tanto, en realidad, somos capaces de obtener una posición precisa aplicando las mismas ecuaciones de la cinemática inversa junto con los nuevos valores.

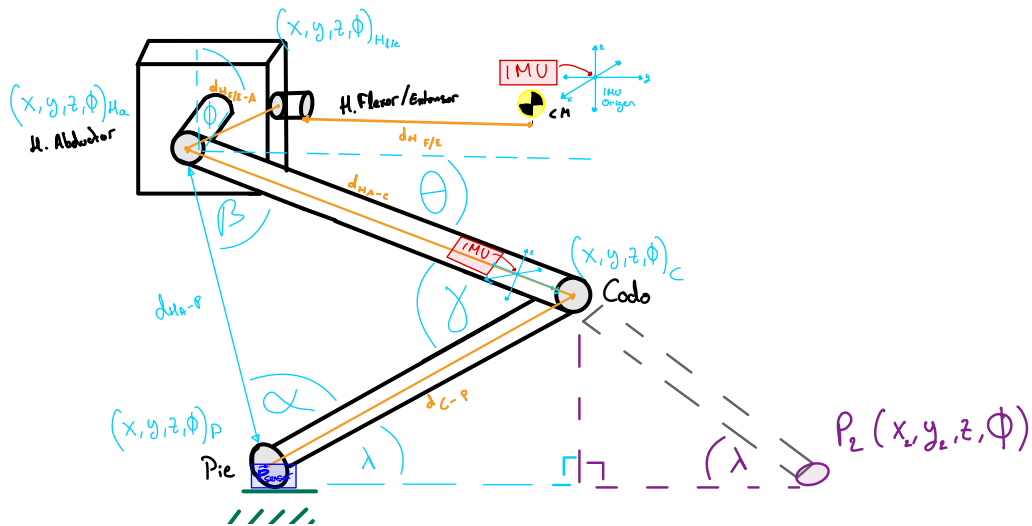


Figura 92. Esquema del sistema IMU-ENCODER posición 2 (Fuente propia).

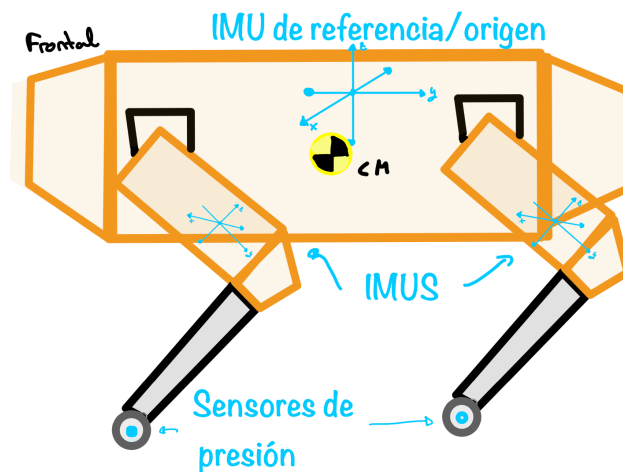


Figura 93. Esquema del sistema IMU-ENCODER sobre el robot (Fuente propia).

Para implementar estas mejoras, será necesario rediseñar las extremidades del robot ya que actualmente solo cuentan con espacio para los 2 cables de los servomotores inferiores. Adicionalmente, se pretende incluir el anillo led en la versión sin *USDAR* ya que permite transmitir información visual de forma directa y sencilla a distancia. Para ello habrá que modificar la tapa superior para que pueda acomodar este sistema.

Por último, teniendo en cuenta que se incorporarán un total de 8 sensores adicionales, 4 de ellos con I2C compartiendo canal, se necesitará un multiplexor de canales I2C para poder gestionar dichas señales. Este componente controlará los sensores inerciales y la pantalla *OLED* 0.96" superior empleando un único puerto I2C. Esto es importante ya que la placa ESP32 solo puede tener 2 canales I2C abiertos simultáneamente. Hasta el momento, el *IMU* principal y la pantalla compartían un canal mientras que la placa de control PCA de los servomotores empleaba su propio canal. En el caso de no emplear este multiplexor, necesitaríamos un mínimo de 3 canales distintos seleccionando la ruta alternativa de los *IMUs*.

### 5.5.2 Elección de los nuevos componentes electrónicos y sensores

Tal y como se ha propuesto en el apartado anterior, se necesitará incluir nuevos sensores y componentes. En el caso de los **sensores inerciales**, se emplearán aquellos del mismo modelo que el actual. Aunque se pudiera pensar que al emplear modelos de otra marca se reducirían los canales I2C, el hecho es que necesitaríamos 3 marcas distintas y se perdería la uniformidad en los datos. Manteniendo el mismo modelo, los posibles errores del sistema son comunes a todos los componentes y el código que interpreta sus valores, reutilizable.

Obtenidos los sensores inerciales, se buscó una **placa multiplexora para I2C** con un mínimo de 4 canales, 2 de 2 parejas de *IMUs* por canal y uno con la pantalla y el *IMU* central. Se decidió optar por la placa TCA9548A 1-to-8 I2C MUX. Esta placa permite una entrada de 3.3 a 5 V y puede gestionar hasta 8 canales. De esta forma es posible mantener cada componente en su canal individual con 2 espacios disponibles para ampliaciones. Otra ventaja es que no requiere modificar la codificación del canal de los *IMUs* y por tanto simplifica tanto el código, el montaje y desmontaje para arreglos y la detección fallos en el sistema.

Por último, en cuanto a los **sensores de presión o fuerza**, se presentan tres opciones principales: resistivos, capacitivos o de efecto Hall:

- **El sensor capacitivo** se uso en la solución para el calibrado automático de la impresora 3D y presenta una buena repetitividad (aunque condicionada a la humedad y temperatura del entorno) para la detección. Otro inconveniente es que debido a su tamaño debería de acoplarse al lateral de la pata y con una alimentación directa de la batería, requiriendo un puente con diodo similar al de la impresora 3D. Sin embargo, el factor descalificatorio serían sus falsas detecciones y distancia de detección mínima. Esto se debe a que por la naturaleza del movimiento. Al colocarlo en la posición de la extremidad podría detectar el propio cuerpo del robot. También era posible detectar el suelo durante la marcha en un estado de transición sin contacto real.
- **El sensor de efecto Hall** no ha sido planteado anteriormente, pero se encuentra presente en algunos encoders industriales. El problema principal de este sensor es que, aunque pueda presentar datos con mucha precisión, es muy sensible y requiere de protecciones especiales que no se pueden garantizar con esta implementación. Además, normalmente emplean discos metálicos sobre los que realizar medidas y al posicionarlo en la punta de apoyo aislada en plástico podría no funcionar.
- **El sensor resistivo** es el más sencillo de los 3 planteados en cuanto a que se trata de un cable entre capas que al ser presionadas aumentan o disminuyen la resistencia (según el modelo). Estos sensores similares a las bandas extensiométricas son flexibles y solo proporcionan lectura tras el contacto, por tanto, son relativamente precisos y no presentan falsos positivos (salvo en averías). Por el peso del robot, se deberían obtener unas de 0 a 1kg (las más pequeñas) y podrían ser encapsuladas por la punta flexible de TPU para estar protegidas del entorno. Aprovechando esto, se podría pasar el cableado por el interior de la pieza protegiéndolo y manteniendo la estética.

### 5.5.3 Diseño de las nuevas piezas del robot

Con las nuevas piezas definidas, el siguiente paso consistió en modificar las piezas existentes y diseñar aquellas que fuesen necesarias para incorporar el nuevo abanico de sensores y componentes.

En primer lugar, se modificó el **circuito principal**, ya que para la tercera fase (v3) se requería incorporar el multiplexor. Al no haber espacio disponible para el mismo en la parte superior, se optó por colocarlo en la parte opuesta al cableado de la batería de la parte inferior (figuras 94 y 95). Se tomaron medidas sobre el circuito anterior y se trasladaron al modelo *CAD*. Adicionalmente a los agujeros para los tornillos y tuercas del multiplexor, se realizaron 3 orificios en la base del circuito para poder pasar los cables de la parte superior al multiplexor de forma similar a como se hace para el cableado de la tensión. Se aprovechó la ocasión para engravar la versión de la placa con *ALFREDog MK3* (*Autonomous Land Farer Robotic Exploration DOG*), corregir sutilmente la elevación de los conectores hembra de los micro USBs y reforzar los puntos de anclaje de los tornillos añadiendo bordes a las bases.

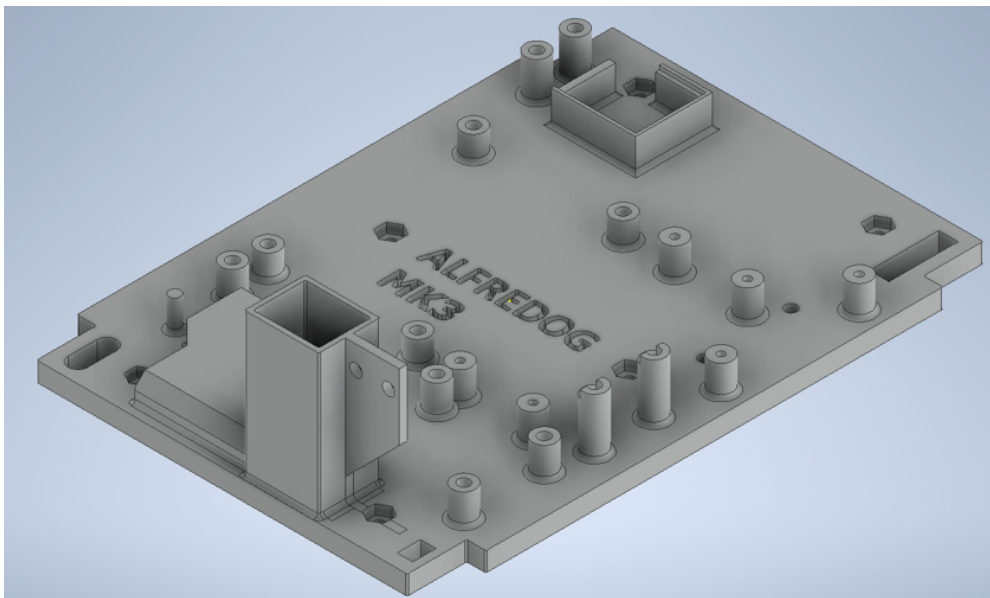


Figura 94. Vista superior del circuito v3 (Fuente propia).

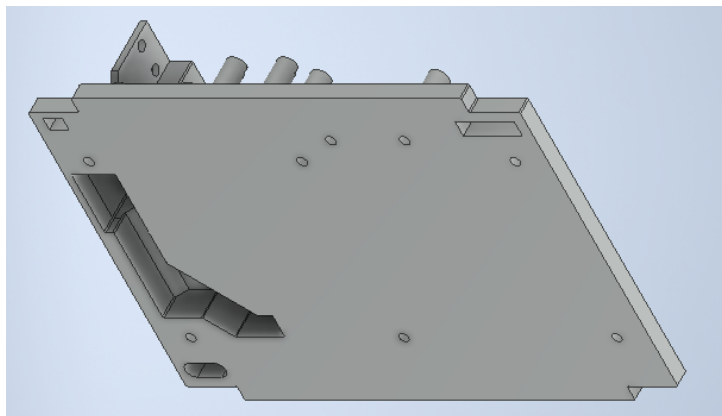


Figura 95. Vista inferior del circuito v3 (Fuente propia).



El siguiente componente en someterse a un rediseño fue el **antebrazo** (*wrist*). El objetivo era crear un canal para los dos cables que conectan los polos del sensor de presión (en realidad, requiere en total 3 cables, esto será explicado en el punto 5.5.5). De esta forma, el sensor quedaría perfectamente pegado a la superficie de la punta y por tanto contenido en la cobertura del pie. La solución fue excavar un canal desde la salida del cable del servomotor hasta la punta. Debido a la flexibilidad de los cables, se apostó en un primer lugar, tras realizar las mediciones adecuadas, por una implementación que mantuviese la mayor rigidez estructural (Figura 96) como se puede ver a continuación:

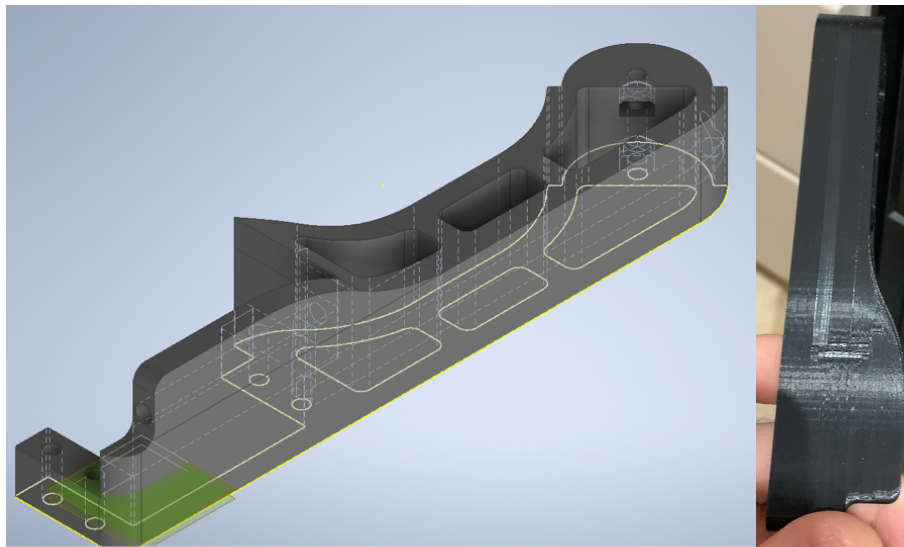


Figura 96. Vista interior del antebrazo v2 (izquierda) y real (derecha) (Fuente propia).

A pesar de que los orificios eran lo suficientemente espaciosos, resultó prácticamente imposible realizar ese giro con un cable y el segundo no lo logró. La solución fue sacrificar algo de rigidez estructural por conveniencia. Para lograr esto se rediseño el sistema de canal con un perfil mucho más ancho en la zona de la curva y con tubos cuadrados para maximizar el volumen interno. El resultado final. (Figura 97) contiene una rampa interna para el cableado. Esta última implementación si cumplía con los requisitos y no requirió modificaciones adicionales:

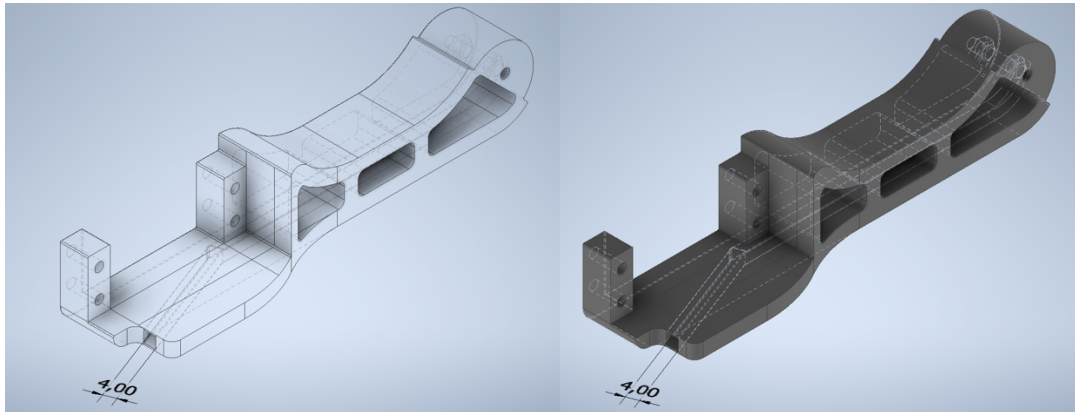


Figura 97. Vista monocroma del antebrazo v3 (izquierda) y simulada con el interior visible (derecha) (Fuente propia).

La siguiente pieza fue el **brazo** del robot (*Limb*), concretamente la tapa inferior que alberga la montura del servomotor. Inicialmente se valoró la creación de una tapa superior que albergase el sensor inercial, pero por el tipo de conexión *Dupont*, fue descartado. Tras realizar una serie de medidas y comprobaciones, se determinó que el sensor cabía en la zona designada para pasar el cable, pero requería un agrandamiento para no impedir que dicha funcionalidad se viese comprometida y alojar el sensor. La solución consistió en agrandar el espacio y generar los orificios para los tornillos y tuercas que sujetarían el sensor. Debido a un fallo en el modelo del sensor (Figura 98), la primera versión no tenía las dimensiones apropiadas, pero fue corregido tras la impresión de prueba (Figura 99). A continuación, se pueden ver las distintas ilustraciones:

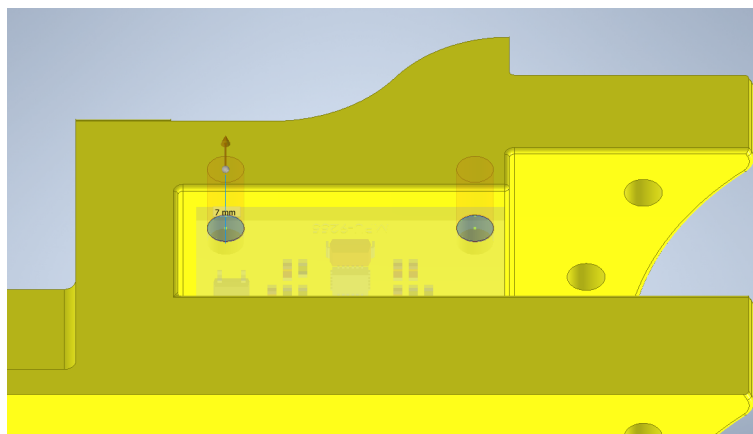


Figura 98. Creación de los canales para los tornillos del brazo v2 (Fuente propia).

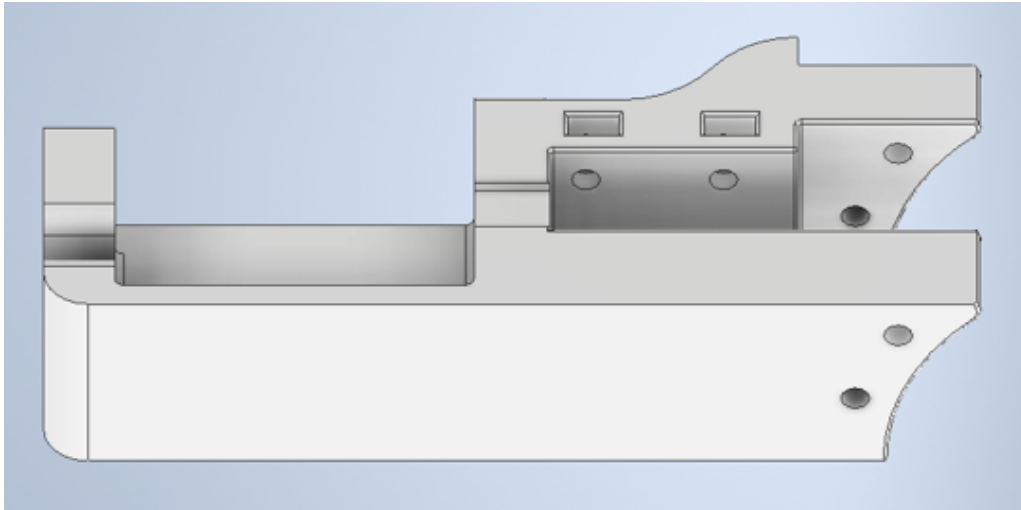


Figura 99. Modelo 3D del brazo v3 (Fuente propia).

Una vez obtenido el sistema, se modificó también el orificio de salida para los cables ya que necesitaríamos pasar 5 cables adicionales a los previstos. Esto también se realizó para la tapa superior, aunque no sería imprimida en pos de agrandar manualmente el orificio de las ya impresas.

Más tarde, haciendo la prueba de ensamblado se comprobó que era imposible pasar dicha cantidad de cables (8 en total). El máximo número de cables era 3-4. Esto se debe a una limitación en el espacio disponible en la tapa superior, la cual de por sí se haya limitada por el grosor. La solución que exigía mínima modificación consistía en generar una tapa para la tapa del brazo. Aunque pueda parecer redundante, en realidad se trata de un **pasador de cables**. Esta tapa con un grosor de 2 capas y un canal para 6 cables se fusionaría, como si de una soldadura se tratase, a la tapa superior. La razón para hacerlo así era una reducción exponencial en la complejidad de la pieza y la reutilización de los recursos ya existentes ya que no cambiaría su funcionalidad. Por tanto, la tapa (Figura 100), imita la geometría inferior y cuenta con una barrea superior para asegurarse de que los cables entran en el tupo de protección. Esta solución se puede ver a continuación:

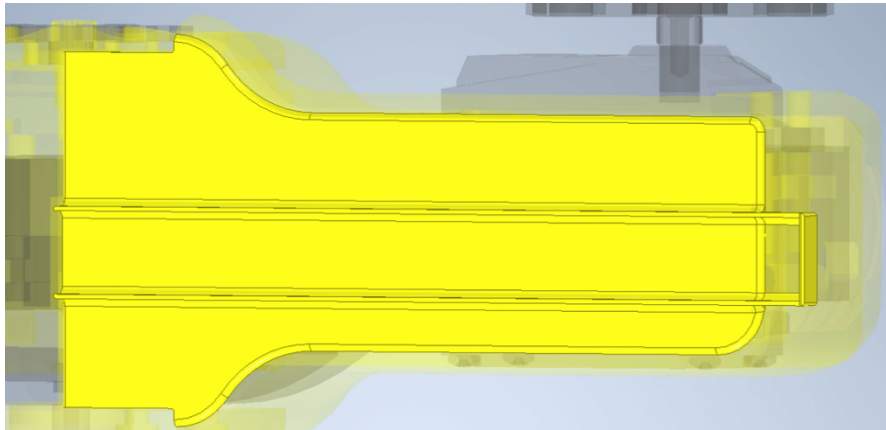


Figura 100. Modelo 3D del pasacables (Fuente propia).

Adicionalmente, se requeriría una **tapa para el pasacables** (Figura 101) esta sería la versión hembra del raíl de los cables y también sería soldada a esta una vez pasados los cables. Con esta implementación se obtuvo una mayor simplicidad de impresión y reutilización manteniendo la misma estética final.

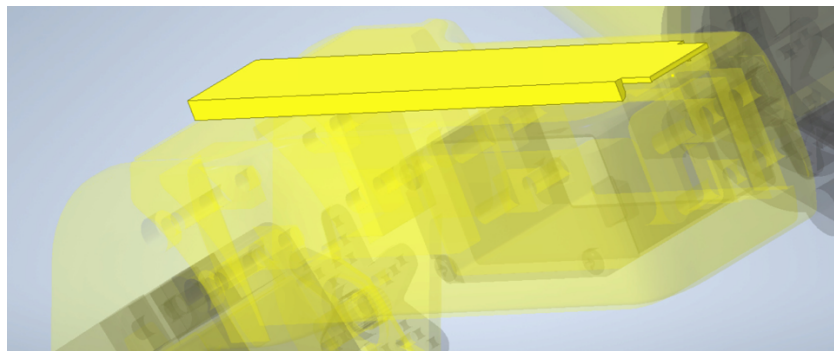


Figura 101. Modelo 3D de la tapa del pasacables (Fuente propia).

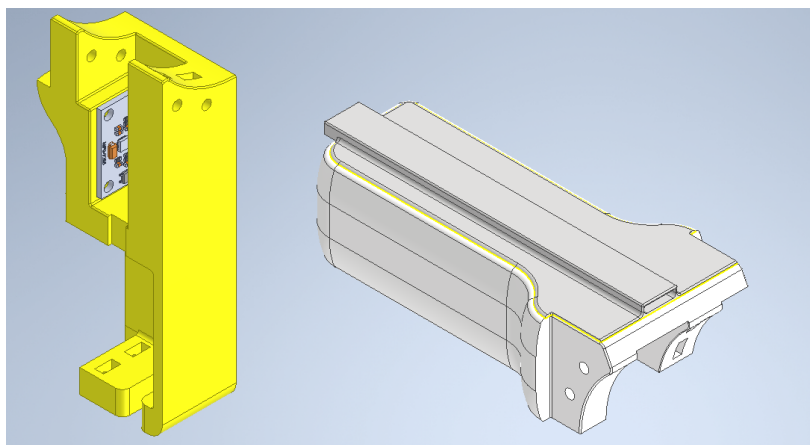


Figura 102. Modelo 3D del brazo v2 del pasacables con su tapa (derecha) (Fuente propia).

Por último, se modificaron los tamaños de los orificios para cables de los soportes de los hombros y del brazo para acomodar el incremento de 6 a 11 cables por pata, así como se incluyeron orificios para la montura del anillo *LED* en la tapa sin *USDAR*. Estas modificaciones no se imprimieron y se realizaron con herramientas manuales. Adicionalmente, se diseñaron unas tapas frontales negras con funcionalidad puramente estética para marcar la versión de la fase 3 (Figura 104).

#### 5.5.4 Ensamblado y comprobación del modelo virtual

Con las nuevas piezas listas, el siguiente paso consistió en la evaluación del modelo mediante el ensamblaje completo de los componentes (figuras 103 a 107). Como se puede ver a continuación, al no modificarse el modelo cinemático y las nuevas piezas encajando correctamente, se pudo validar el modelo de la fase 3. Esto es tanto para la versión con *USDAR* como para la versión básica.

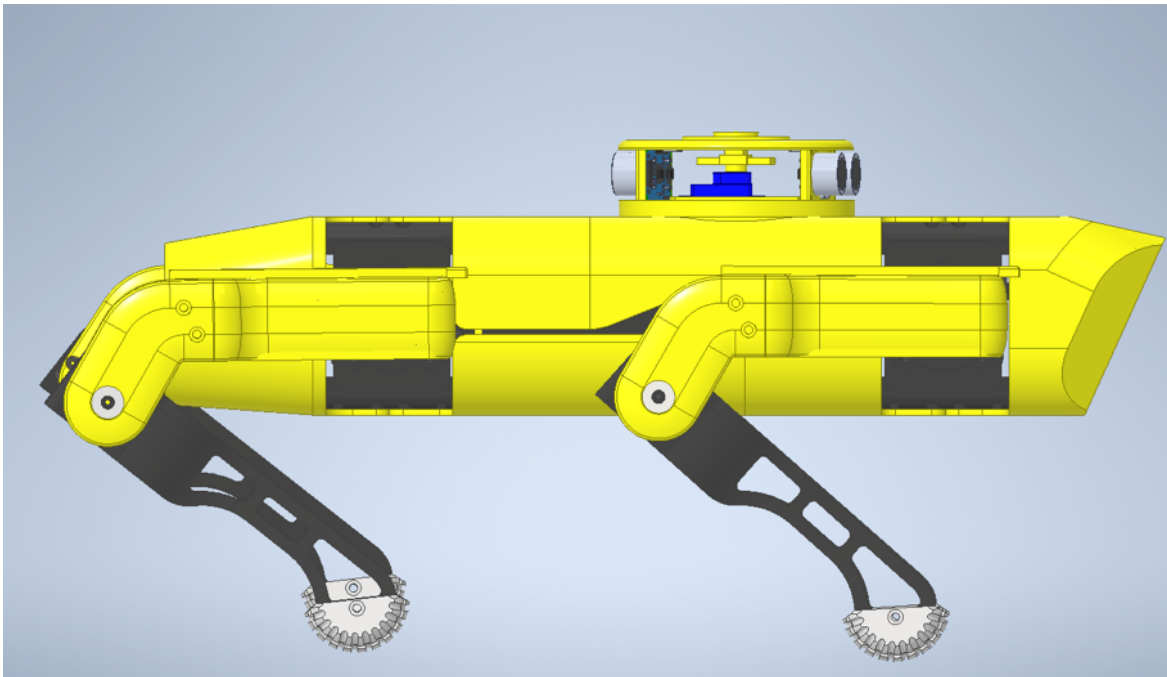


Figura 103. Modelo 3D del robot en la fase 3 con *USDAR* vista lateral (Fuente propia).

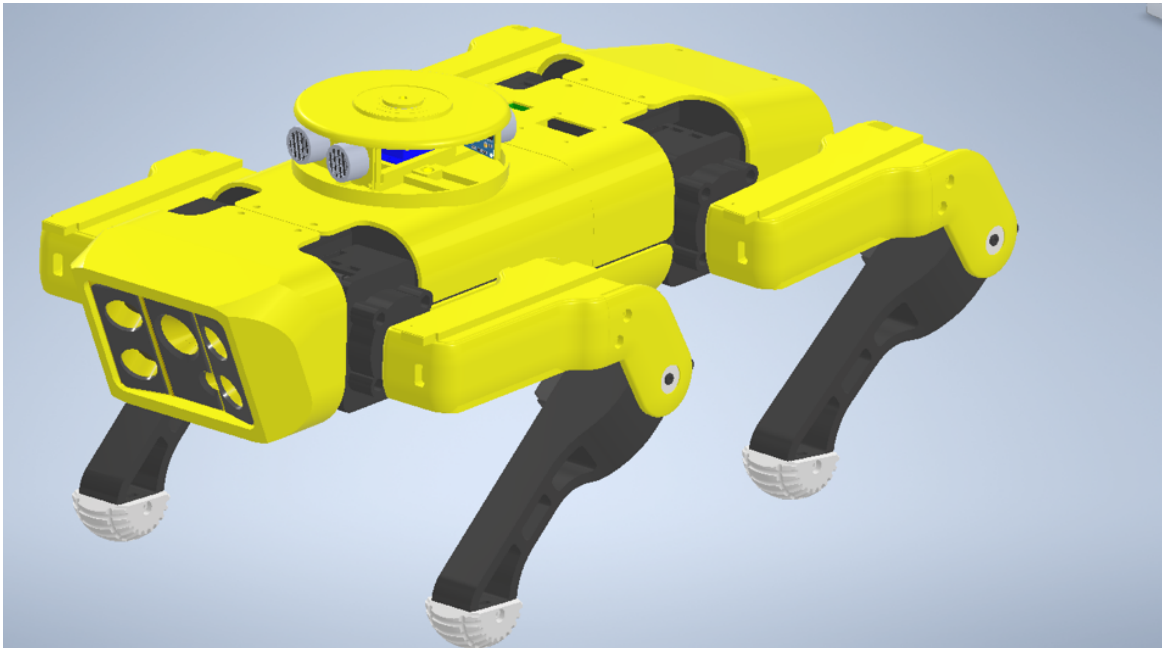


Figura 104. Modelo 3D del robot en la fase 3 con *USDAR* vista frontal izquierda (Fuente propia).

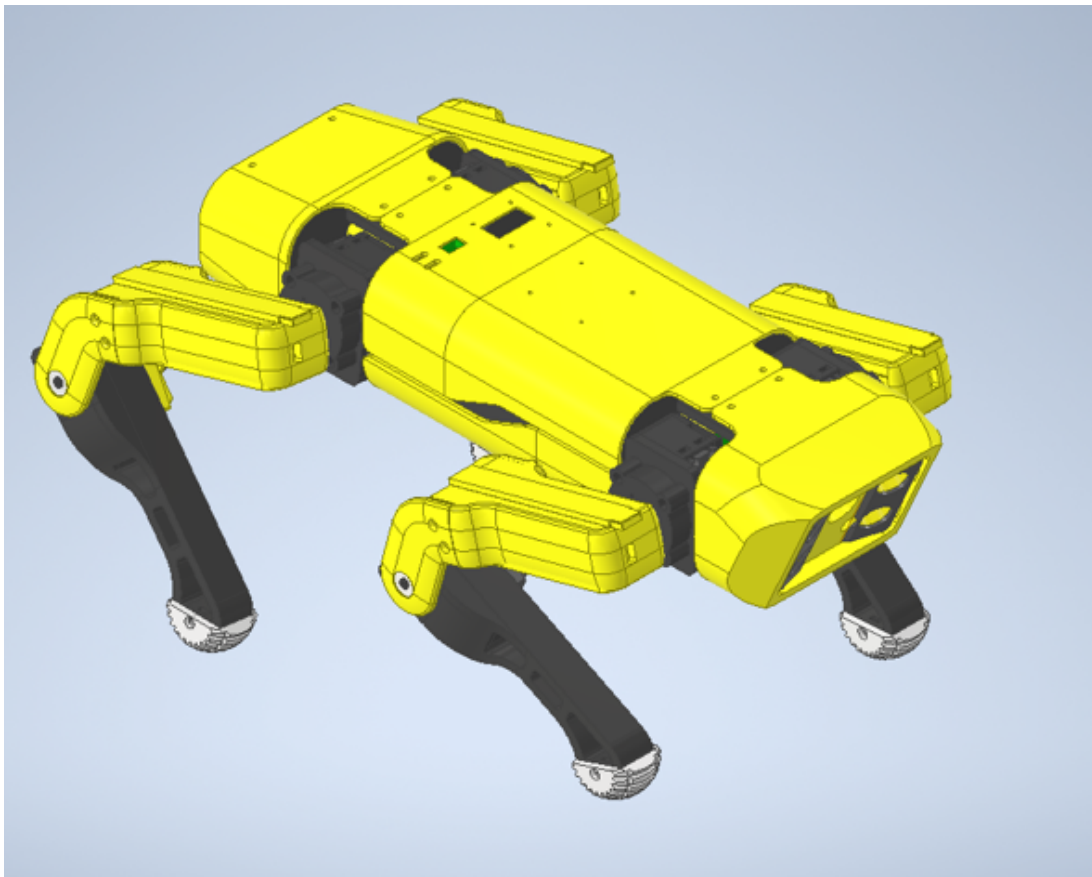


Figura 105. Modelo 3D del robot en la fase 3 básica con vista frontal derecha (Fuente propia).



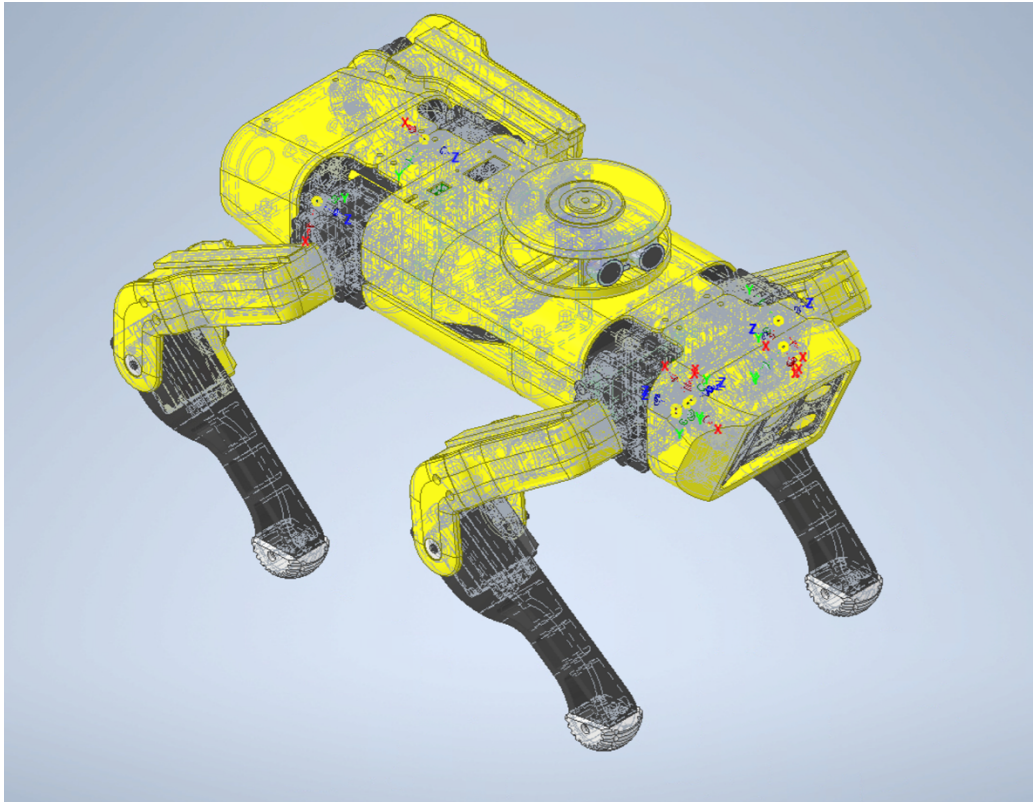


Figura 106. Modelo 3D del robot en la fase 3 con *USDAR* con vista frontal derecha de las aristas internas (Fuente propia).

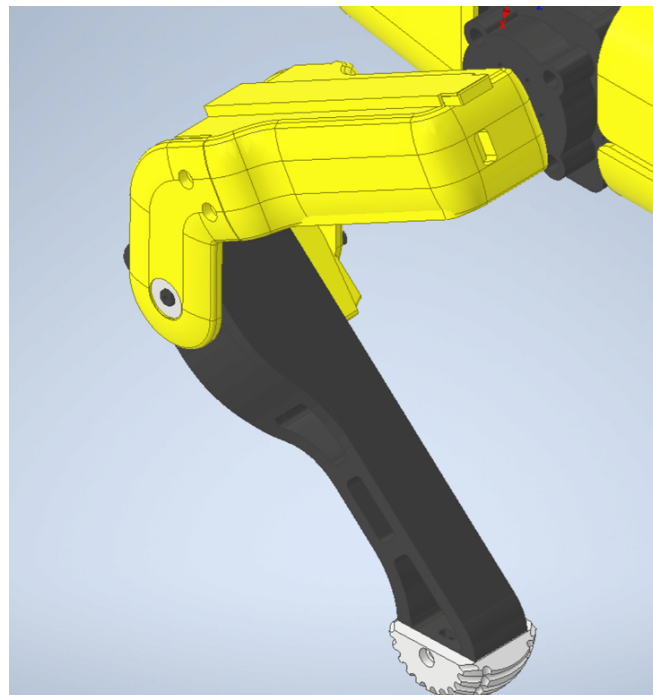


Figura 107. Modelo 3D de una pata derecha del robot en la fase 3 (Fuente propia).

### 5.5.5 Revisión del esquema electrónico

Con la adición de los nuevos sensores, era necesario modificar el esquema electrónico. Como se mencionó en el apartado 5.5.3, se añadieron un total de 5 cables por pata. Esto es debido a que cada *IMU* requiere 4 cables (Vcc,GND,SDA y SCL) y el sensor de presión 3 (Vcc, GND, S). La solución para minimizar los cables, que por separado sumarían 7, consistió en alimentar los sensores con el mismo cableado. La técnica concreta es explicada en mayor profundidad en el [vídeo de la construcción](#), pero en resumidas cuentas consistía en emplear una resistencia de 100 K $\Omega$  en la salida del polo negativo del sensor de presión. Esto generaría 2 cables: S (directo) y GND (tras resistencia). A GND le acoplaríamos el cable de salida a la fuente y el de entrada al IMU, y el positivo lo dividiríamos directamente. Esto no afectaba al funcionamiento del circuito ni de los sensores ya que estarían en paralelo. Así reduciríamos en 3 el número de cables.

Seguidamente, tendríamos que pasar por la barriga del robot esos nuevos cables provenientes de las patas para la nueva placa multiplexora I2C y los cables de la pantalla, *IMU* de referencia, control y alimentación del MUX desde la parte superior del circuito. Los demás componentes se mantienen en el mismo punto, aunque fue necesario cambiar de la alimentación inicial de 5V a 3.3V por el ADC de la ESP32 (punto 5.6.4).

Para esta fase hay un total de 6 versiones del circuito en base a pequeñas variaciones detectadas y cambios en el diseño debido a problemas con la placa. Esto es explicado en más detalle en el punto 5.6.4. A continuación se puede apreciar en la Figura108, la última implementación de este, ya que la diferencia con las anteriores es el cambio en la alimentación y algunos puertos GPIOs.

A continuación, se encuentra la leyenda de colores para los cables de la fase 3 (contiene una *Raspberry Pi ZW* y una fuente eterna de alimentación):

- Negro y marrón: tierra o GND.
- Magenta: polo positivo de la batería 8.4V.
- Cian oscuro: polo positivo de 6.5V.
- Morado: polo positivo de 5V.
- Rojo: polo positivo de 3.3V.
- Verde: I2C SDA.
- Amarillo: I2C SCL.
- Verde: Echo US



- Amarillo: Trigo US
- Otros: cables empleados para señales digitales.
- Resistencias: 100 k $\Omega$

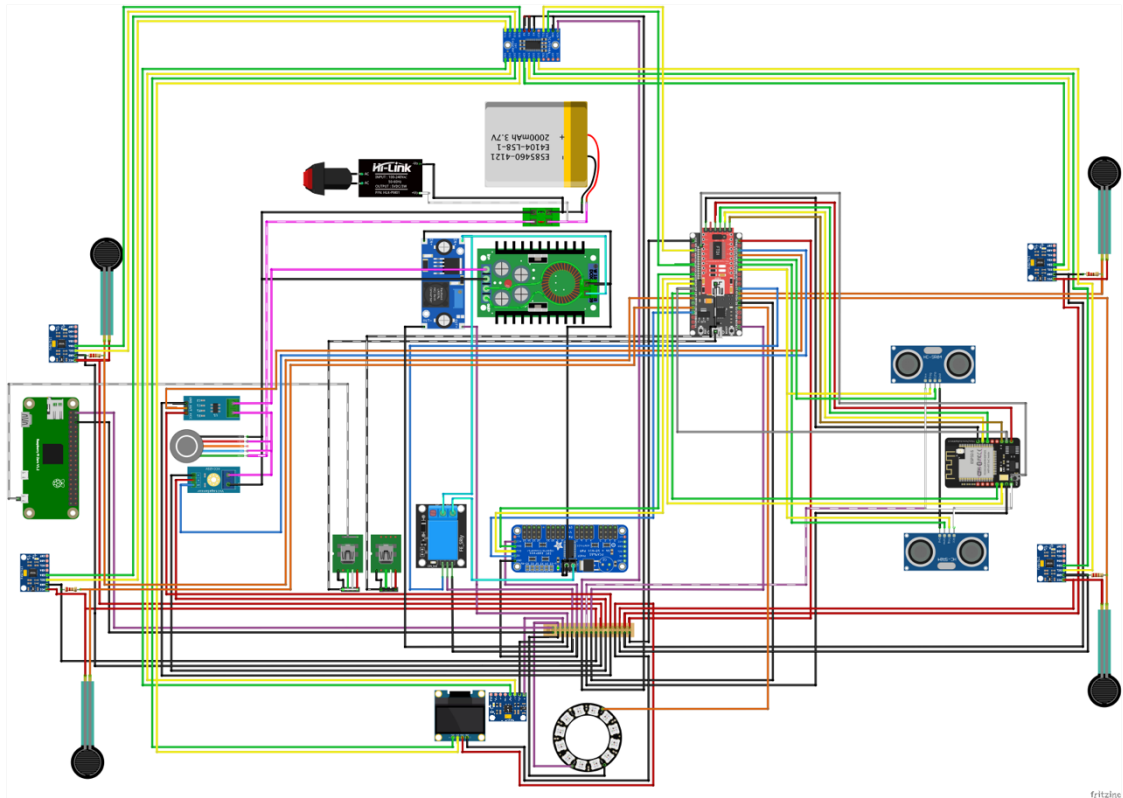


Figura 108. Diagrama final de conexiones (circuito\_v9) de la fase 3 (Fuente propia).

### 5.5.6 Elaboración de los archivos de impresión y revisión de las tablas de costes

Con todos los componentes verificados, se procedió a la elaboración de los archivos de impresión y la elaboración de las tablas de costes temporales y económicos para el cambio de fase. Estas tablas se encuentran actualizadas con los cambios y problemas surgidos a lo largo de esta tercera fase suponiendo el coste final de su desarrollo. Las tablas 8 y 9 se encuentran a continuación:

ALFREDog - FASE 3 - COSTES DE IMPRESIÓN						
Hombros y soportes						
Cantidad	Nombre de la pieza	Relleno	Duración	Filamento	Total de la duración	Total de filamento
Miembros						
Cantidad	Nombre de la pieza	Relleno	Duración	Filamento	Total de la duración	Total de filamento
5x	Base inferior del brazo	20%	5h 10min	30 g	20h 40min	150 g
4x	Cubierta de brazo	20%	1h 30min	12 g	8h 28min	48 g
4x	Tapa de la cubierta del brazo	20%	29min	5 g	1h 56min	20 g
4x	Servohorns Bstar	100%	31min	7 g	2h 4min	28 g
					<b>Total</b>	<b>246 g</b>
Pies						
Cantidad	Nombre de la pieza	Relleno	Duración	Filamento	Total de la duración	Total de filamento
8x	Pie con hueco	20%	4h	27 g	16h 40min	216 g
					<b>Total</b>	<b>216 g</b>
Cubiertas del chasis y circuito						
Cantidad	Nombre de la pieza	Relleno	Duración	Filamento	Total de la duración	Total de filamento
1x	Cubierta superior trasera	100%	2h 21min	28 g	2h 21min	28 g
1x	Circuito v3	20%	8h 14min	60 g	8h 14min	60 g
					<b>Total</b>	<b>88 g</b>
					<b>Total PLA Negro</b>	216
					<b>Total PLA Amarillo</b>	334
					<b>Total de filamento</b>	550

Tabla 8. Costes de tiempo y materiales de impresión para la fase 3 (Fuente propia).

ALFREDog - Total de costes de los materiales - Fase 3					
Nombre	Modelo	Cantidad	Precio €	Total €	
Microcontrolador	ESP32-DevKitC	2	7	14	
Pantalla	0.96" OLED I2C Display	2	8	16	
Sensores de distancia	HC-SR04 Ultrasonic Sensor	1	3	3	
Sensor inercial	GY-521 Gyroscope and Accelerometer	5	4	20	
Controladora de motores	PCA9685 16Channel 12Bit PWM Board	1	14	14	
Conexión externa USB	Micro-USB to DIP Adapter	2	6	12	
Mando	Logitech Gamepad F310	1	30	30	
Multiplexor I2C	Placa TCA9548A 1-A-8 I2C 8	2	7	14	

Controlador principal	Raspberry Pi zero W	1	30	30
Shield de USB Rpi	Conector para usbs de la pi zero	1	15	15
Plástico adicional	Bobina de 1kg de pla	1	20	20
Sensores de presión	1kg resistive sensors	4	3	12
Total	No incluye el gasto de luz			200

Tabla 9. Costes de los materiales y componentes para la fase 3 (Fuente propia).

### 5.5.7 Impresión de las piezas estructurales

Una vez dispuestos los modelos, se comenzó la fase de impresión y prototipado. En esta fase hubo un problema con la impresora ya que los ejes habían resultado dañados por un fallo de uno de los motores y este a su vez había provocado que el extrusor se sobre saturase. Esto fue percibido tras la impresión de 3 piezas con algunas capas sin material y otras poco cohesionadas (Figura 109) lo que supuso un retraso de 2 semanas y media mientras se reparaban las piezas, aplicaba mantenimiento y se imprimían de nuevo bajo supervisión. Esta tercera tanda de impresiones duró una semana y media con un total aproximado de 60 h de impresión contando los defectos. Por lo que la elaboración de las piezas supuso 1 mes de impresión y reparación de la impresora. Solo una de las piezas se pudo reparar con soldadura, pero el resto quebraron bajo el ensayo a compresión (Figura 110).

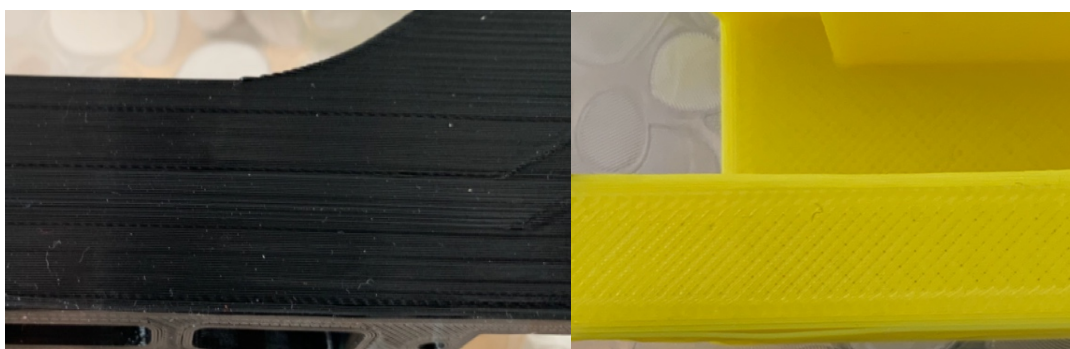


Figura 109. Vista lateral del defecto de capa del antebrazo (izquierda) y del brazo inferior (derecha) (Fuente propia).



Figura 110. Vista superior del defecto de capa del antebrazo quebrado a tensión (Fuente propia).

## 5.6 Construcción y verificación: Fase 3

### 5.6.1 Preparación de las piezas preexistentes

Una vez impresas las nuevas piezas para la fase 3 (Figura 111) se procedió al desmontaje y preparación del resto de componentes que requerían mecanizado para agrandar sus orificios para cables.

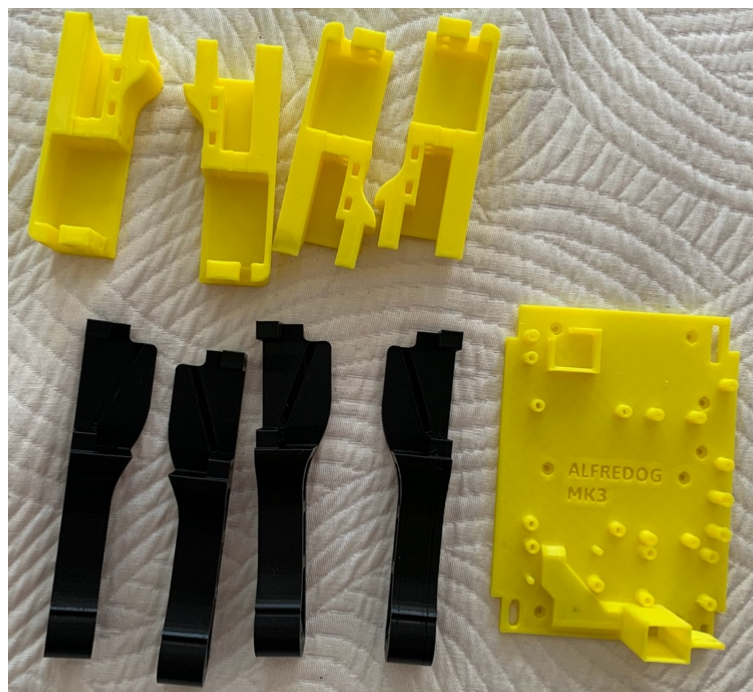


Figura 111. Vista superior de las piezas impresas para la fase 3 (sin contar las tapas no visibles) (Fuente propia).



Figura 112. Vista superior de las piezas impresas para la fase 3 del brazo (Fuente propia).

Sobre estos componentes se realizó una ampliación del diámetro de sus orificios empleando una amoladora y un soldador. El procedimiento consistió en marcar con un rotulador permanente la zona medida que se pretendía sustituir (Figura 113), fundir esa zona para mover el material (lo cual aumenta la resistencia local al concentrar más plástico) y pulir los sobrantes anómalos a la geometría.



Figura 113. Preparación para la ampliación del orificio del brazo superior (Fuente propia).



Preparados los hombros y los componentes del brazo, se decidió hacer un cambio de última hora y sustituir el terminal de conexiones casero que se diseñó para la fase 1, por uno casero montado sobre una *protoboard* en lugar de PLA (Figura 114). Esto era necesario debido a la ampliación de pines profiláctica que luego resultaría indispensable en el punto 5.7.3. Para ello, se recortó una *protoboard* con una radial. A esta, se le soldaron dos filas de 18 pines separados por tensiones (3.3 y 5 V) empleando distintos cabezales de soldadura. La comparativa se puede ver en la siguiente imagen:



Figura 114. Terminal de conexiones de alimentación sobre protoboard (izquierda) y original de la fase 1 y 2 (derecha) (Fuente propia).

### 5.6.2 Preparación del nuevo cableado eléctrico interno

Esta fue la parte más complicada de todo el proceso ya que sería la que posteriormente supusiese un retraso de 3 semanas solucionando errores derivados de la implementación (expandido en las conclusiones del [capítulo 6](#)). Con la adición de los nuevos componentes, fue necesario preparar un gran número de cables. A continuación, se detallarán los necesarios cuyo proceso de construcción será expandido tanto en el vídeo de la construcción [en la nube](#) como en el punto 5.6.5.

Para las **piernas** se dispusieron un total de 17 segmentos de cables de 10cm que serían a su vez empleados para 2 líneas de 4 y 3 líneas de 3. El objetivo era poder, en caso de desmontar el robot, hacerlo por secciones. En el extremo tendríamos dos cables conectados al sensor de presión de los cuales soldaríamos una extensión al VCC y una resistencia de 100 k $\Omega$  y 2 cables al GND/S. En el caso de los I2C, estos irían directamente desde el IMU. Para estos, se prepararon cajetillas Dupont acorde al número de cables.

En el caso de la **pantalla** y el **anillo LED** fue necesario soldarle al segundo los terminales y preparar para el primero un cableado partido ya que la señal I2C viajaría al multiplexor en la parte inferior del circuito mientras que la tensión

permanecería arriba. Debido al hecho de que ambos componentes están atornillados a la tapa y con el fin de que esta sea removible, el cableado del anillo cuanta con un terminal macho y la pantalla se puede desconectar directamente de sus Dupont. Esto se puede ver a continuación en la Figura 115 aunque posteriormente sería optimizado y ordenado:

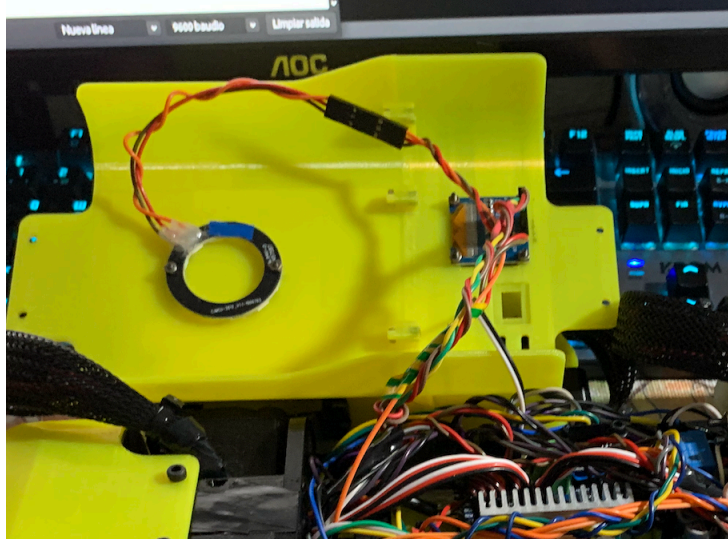


Figura 115. Conexiones del anillo LED y la Pantalla (Fuente propia).

Para el **multiplexor I2C** fue necesario preparar un conector especial ya que para establecer una dirección I2C hay que programársela empleando un código de 3 bits. Para darle la dirección 0x70 se tuvieron que unir los tres terminales de selección a tierra como se puede ver en la Figura 116. Adicionalmente, se tuvieron que preparar los cables de alimentación y señales I2C para su control dejando espacio para los 6 pares I2C que deberían de conectarse más adelante.

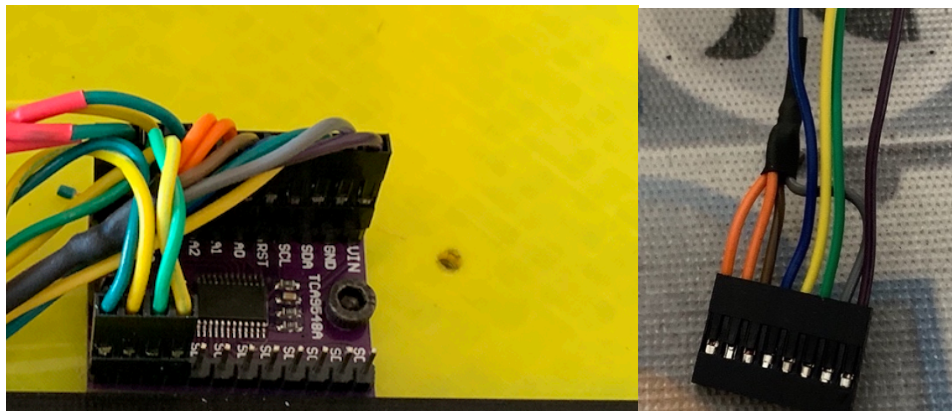


Figura 116. Conexiones del multiplexor I2C (Fuente propia).

Para el **sistema de alimentación** se rehízo la alimentación al *hub* de alimentación para que esta fuese suministrada por la parte inferior (Figura 117) y se rehízo la de los sensores de alimentación y el pulsador por unas conexiones más limpias y largas aplicando habilidades de soldaduras adquiridas con la práctica (Figura 118). Posteriormente, se produjo un cortocircuito y pequeño incendio durante las pruebas. Fue provocado por un falló del relé en modo intermitente que provocó daños en la PCA cortocircuitando su entrada de 6.5 V y fundió el transformador LM de 5V que a su vez dañó todos los componentes electrónicos conectados a su salida. El resultado de esto fue que se tuvo que cambiar el cableado de alimentación y sustituir los componentes dañados suponiendo una semana de retraso rehaciendo y adquiriendo el material. Pues el fallo en la PCA no apareció hasta repararlo todos unos días después dañando el *ADC* de la nueva ESP32 (Figura 119). Esta es la parte que sin duda más ha afectado al desarrollo al suponer un retraso de 2 semanas y la pérdida del 60% de los componentes electrónicos.

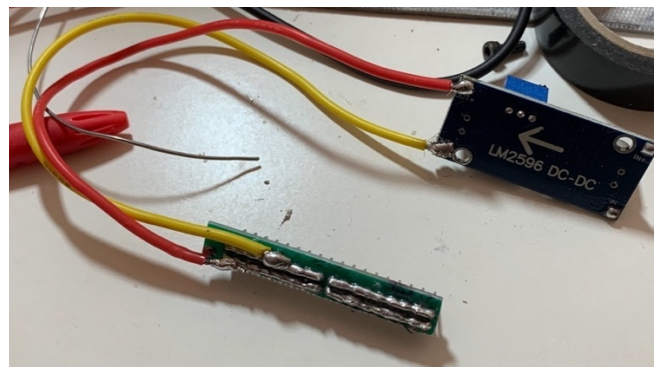


Figura 117. Nuevas conexiones del LM al *hub* de tensiones (Fuente propia).

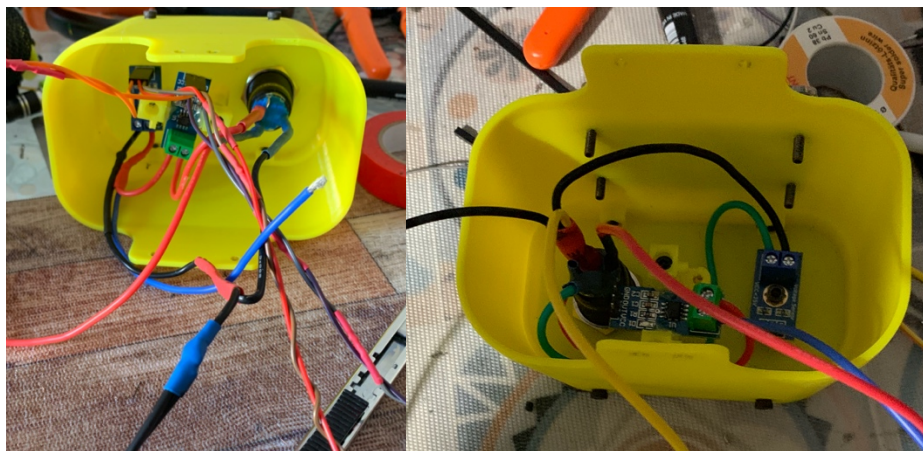


Figura 118. Comparativa del nuevo cableado de alimentación a la izquierda el antiguo y a la derecha el modificado en la fase 3 (Fuente propia).



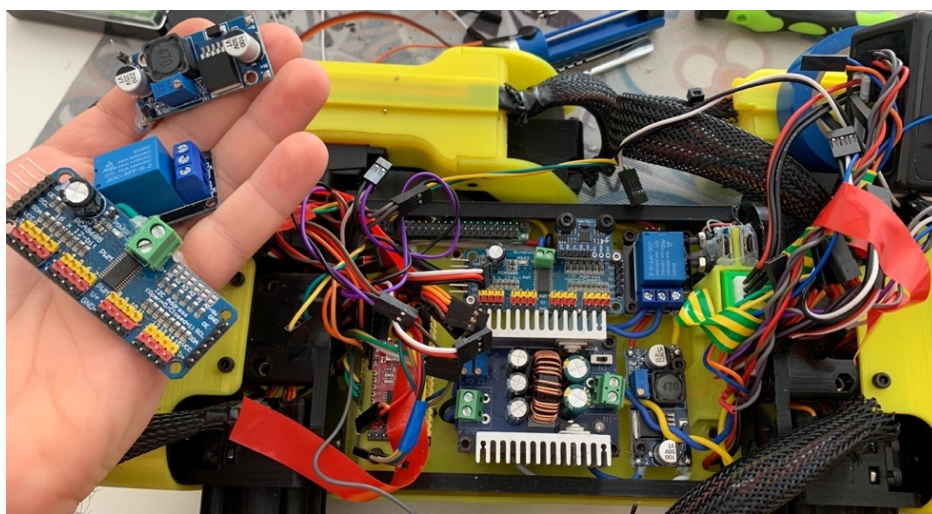


Figura 119. Circuito tras la sustitución de los componentes que provocaron los 2 cortocircuitos de la fase 3 (Fuente propia).

Por último, el ajuste general de los **componentes principales** en este caso, se trenzaron los cables para proporcionar mayor resistencia y se agruparon en conectores múltiples de Dupont hembra para que el montaje y desmontaje fuese más sencillo y las conexiones fuesen reforzadas.

### 5.6.3 Elaboración de códigos de testeo y pruebas de funcionamiento

Al realizar cambios y añadir nuevos componentes, fue necesario elaborar nuevos códigos de prueba. En este caso, se elaboró un código para verificar cada uno de los nuevos componentes:

- **Sensores de presión:** el código de los sensores de presión se basa en el análisis por parte del *ADC* de la tensión de entrada como una medida directa de la presión ejercida. Esto significa que cuando no se ejerce ninguna fuerza sobre el sensor, el *ADC* no recibirá señal o tensión de entrada y, para una fuerza de más de 9,8 N, recibirá una tensión igual a la de entrada del sensor. El *ADC* nos devolverá un valor de 0 a 4095 directamente proporcional a la fuerza ejercida. Para verificarlos podemos usar o 3.3 o 5 V pero con la ESP32 es fundamental usar 3.3 V ya que de usar 5V saturaremos el *ADC* corrompiendo sus futuras medidas y a la larga se podría llegar a dañar la placa. Los vídeos de las pruebas de presión se encuentran compartidos en la [carpeta en la nube](#). A continuación, en la Figura 120, se puede ver una imagen de su ejecución:

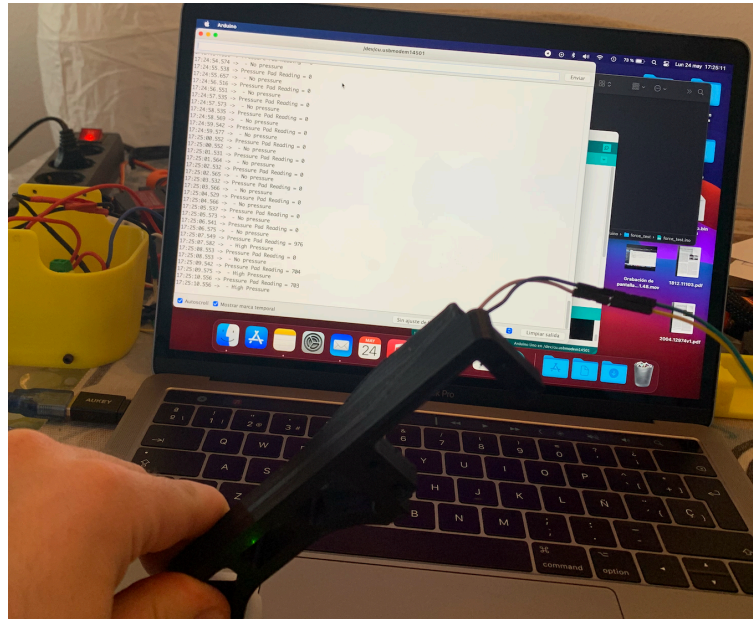


Figura 120. Prueba de la lectura del sensor de presión instalado en la pata (Fuente propia).

- **Multiplexor TCA9548A 1-to-8 I2C:** para probar este multiplexor se elaboró un código a partir del ejemplo dispuesto en su librería y se utilizó para a través de su selector, realizar mediciones sobre un sensor inercial (Figura 121) como se puede ver a continuación:

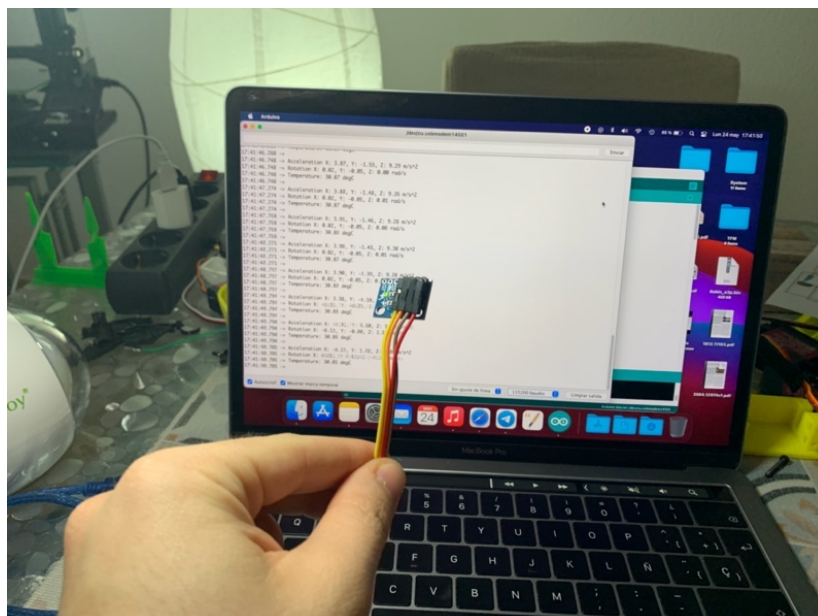


Figura 121. Prueba de la lectura del sensor de inercial mediante el multiplexor (Fuente propia).

### 5.6.4 Calibrado de los motores

Tras tener preparados los sensores, los últimos componentes restantes y cruciales para la elaboración del robot son los servomotores. Estos han sido empleados hasta el momento con funciones de ángulos enteros sin calibrado ya que hasta este momento la precisión no ha sido necesaria. Sin embargo, hay que tener en cuenta que estos servos tienen un ángulo de  $195^\circ$  efectivos y por tanto es necesario conocer los valores exactos de la señal PWM para cada punto. El procedimiento consiste en ir aumentando progresivamente el valor de la señal hasta detectar movimiento, medir  $180^\circ$  y anotar cuando se detiene. Esto nos debería de dar 3 valores: PWM de inicio ( $0^\circ$ ), PWM a  $180^\circ$  y PWM final ( $195^\circ$ ). Tras realizar esta comprobación para todos los servomotores (visibles en el [vídeo de construcción](#)), se observó que con un pulso de  $79 \mu s$  el motor se paraba en  $0^\circ$ , con  $468 \mu s$  alcanzaba los  $180^\circ$  y a los  $502 \mu s$  se detenía. Llegado este momento era necesario obtener la relación entre  $\mu s$  y grados. El objetivo es generar una región entre el inicio posible del movimiento y el movimiento llamada *buffer*. El procedimiento matemático se puede ver en la Figura 122 a continuación:

CÁLCULO PWM (PULSE WIDTH MODULATION)

$$79 \text{ PWM} = 0^\circ$$

$$502 \text{ PWM} = 195^\circ$$

$$502 \text{ PWM} - 79 \text{ PWM} = 423 \text{ PWM ÚTIL}$$

$$\frac{\text{PWM}}{\text{ángulo}} = \frac{423 \text{ PWM ÚTIL}}{195^\circ} = 2.17 \frac{\mu}{\text{grados}} \rightarrow 1^\circ = 2.17 \frac{\mu}{\text{grados}}$$

$$0^\circ = 2.17 \frac{\text{PWM}}{\text{grados}} \cdot 7.5^\circ$$

$$0^\circ \text{ NUEVO} = \left( 2.17 \frac{\mu}{\text{grados}} \cdot 7.5^\circ \right) + 79 \text{ PWM} = 95 \text{ PWM}$$

→ Teniendo en cuenta el BUFFER



Cálculo ángulos clave

Fórmula:  $X^\circ \text{ NUEVO} = \left( 2.17 \frac{\mu}{\text{grados}} \cdot X^\circ \right) + 95 \text{ PWM}$

X = ángulo clave que se quiere calcular

$$0^\circ = 95^\circ$$

$$60^\circ = 225^\circ$$

$$90^\circ = 290^\circ$$

$$120^\circ = 355^\circ$$

$$180^\circ = 485^\circ$$

Figura 122. Cálculo de la señal PWM para el calibrado de los servomotores (Fuente propia).

Con el *buffer* de  $7.5^\circ$  al inicio y al final se es capaz de mantener la movilidad en los  $180^\circ$  pero con un rango adicional de seguridad para garantizar que los motores siempre se encuentran en una zona controlable. El factor

resultante es de  $2.17 \mu\text{s}^\circ$ . Por tanto, nuestro nuevo valor para  $0^\circ$  será  $95 \mu\text{s}$  y para  $180^\circ$   $485 \mu\text{s}$ . Con estos valores, ya podremos tener un control completo del ángulo al que deseamos enviar al robot con un margen adicional de seguridad.

### 5.6.5 Montaje del robot

Tal y como se ha podido observar en los puntos anteriores, se verificaron los nuevos componentes añadidos a la tercera fase además de los previos antes del montaje del robot (visibles en el [vídeo de construcción](#)). A continuación, se detallarán los pasos principales y algunas de las decisiones tomadas de forma general pues se pretende resumir 2 semanas de trabajo de construcción.

- **Montaje del antebrazo (x4):** en primer lugar se soldaron y fijaron los terminales de dos cables al sensor de presión y se pasaron estos por el interior del antebrazo hasta el límite físico del sensor. Seguidamente se pegó este a la superficie (Figura 123). Tras hacer esto, se alargaron dichos cables y se condujeron a través de la malla inferior junto con el cable del servomotor. Hecho esto, sujetaron todas las piezas de la pata inferior con la tornillería correspondiente. Y se prepararon los cables para las 5 salidas (Figura 124) al robot correspondientes a las entradas del sensor inercial y de presión. Ambas figuras son visibles a continuación:

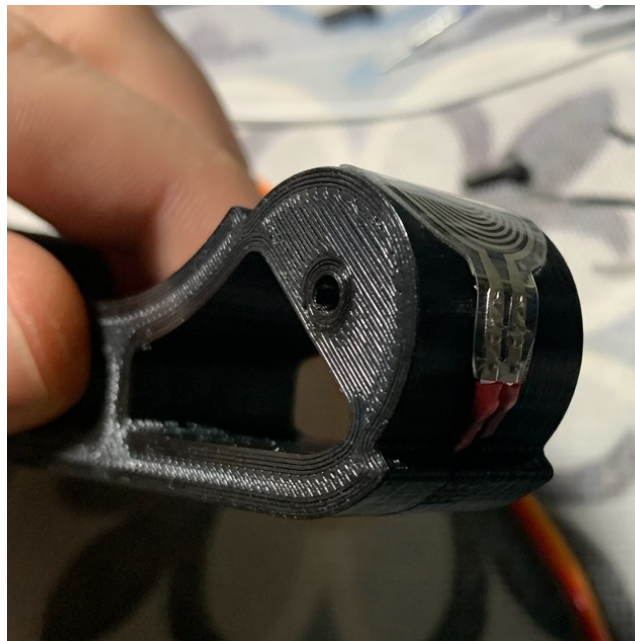


Figura 123. Colocación del sensor de presión (Fuente propia).



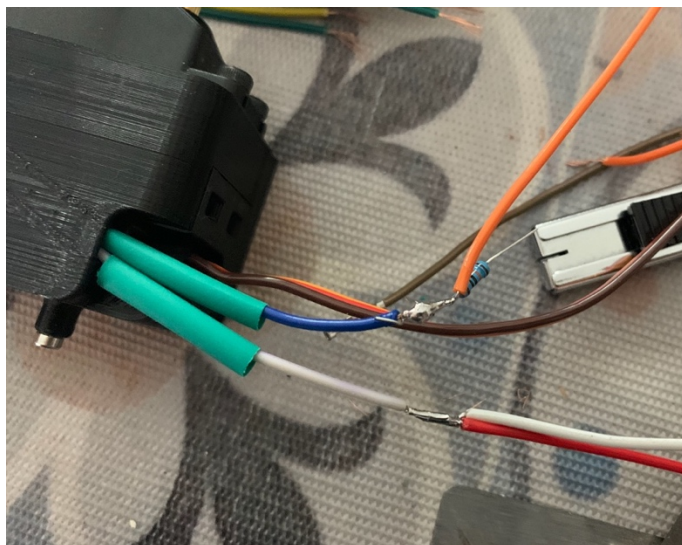


Figura 124. Preparación del cableado para la siguiente articulación (Fuente propia).

- **Montaje del brazo (x4):** el primer paso para el montaje del brazo fue instalar el sensor inercial (Figura 125) una vez hecho esto, se terminó de preparar el cableado para los sensores. El siguiente paso consistió en soldarle a las tapas superiores los pasacables diseñados anteriormente (Figura 126) y alargar un segmento más todos los cables para evitar soldaduras en el tramo interno del brazo. Por último y se pasaron todos los cables y cerraron las tapas soldando en última instancia la última tapa (figuras 128 y 129) instalando los motores en los ángulos correctos. En esta parte una tuerca se pasó de rosca dañando la vía interna en la que se sujetaba lo que resultó en la necesidad de destruir ese brazo al completo para recuperar el motor, cableado, sensores y tornillería (Figura 130). Las imágenes pueden verse a continuación:

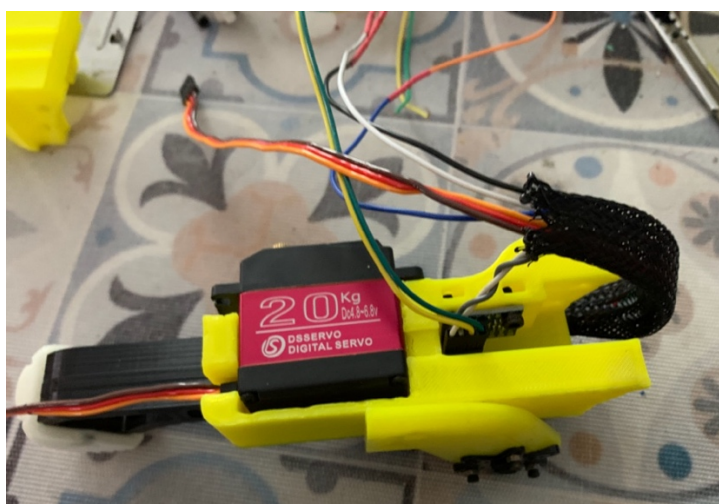


Figura 125. Instalación y conexión del sensor inercial (Fuente propia).



Figura 126. Instantes antes de la soldadura por fusión del pasacables (Fuente propia).

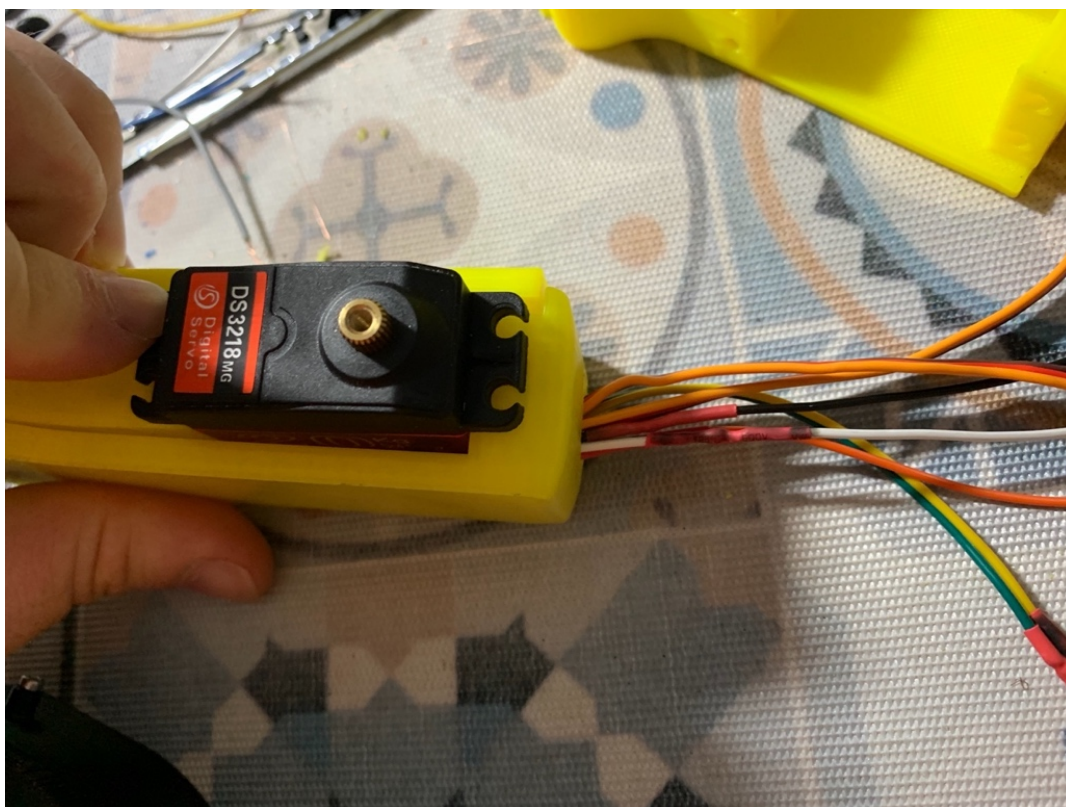


Figura 127. Verificación de la necesidad del pasacables (Fuente propia).





Figura 128. Preparación del cableado para la siguiente articulación y la fijación de la tapa del pasador de cables (Fuente propia).



Figura 129. Construcción de las patas finalizada (Fuente propia).



Figura 130. *Rapid Scheduled Disassembly* del brazo para rescatar los componentes (Fuente propia).

- **Montaje del hombro abductor (x4):** esta es la parte más sencilla y compleja del proceso al mismo tiempo ya que la posición de anclaje del servomotor al hombro debe de estar a  $90^\circ$  del servomotor y  $60/120^\circ$  con respecto al eje del hombro. Esto se puede ver ilustrado en la Figura 131. La solución fue recortar una plantilla (Figura 132) y tratar de realizar el mejor ajuste posible y compensar por *software* variaciones de  $\pm\mu s$  para el PWM.

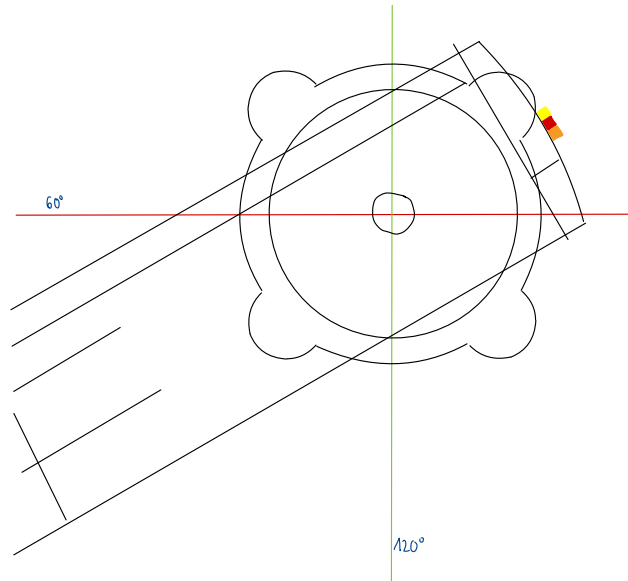


Figura 131. Patrón base para el montaje del hombro abductor (Fuente propia).

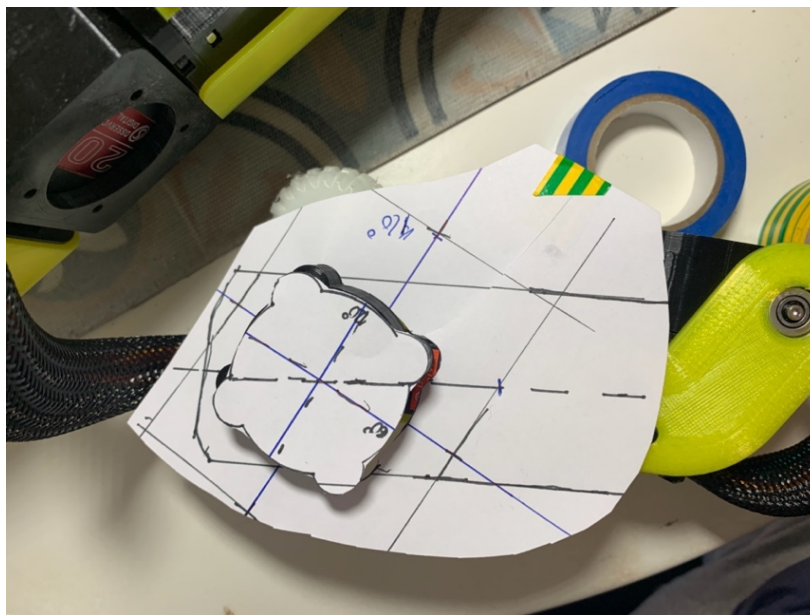


Figura 132. Patrón base para el montaje del hombro abductor en uso (Fuente propia).



- **Montaje del hombro flexor/extensor (x4):** esta construcción se realiza previamente cuando se cambia el circuito ya que solo es necesario colocar los servomotores a 90° y cuadrarlos paralelos al marco del hombro. En la siguiente imagen (Figura 133) se pueden ver ya instalados en el marco y en otras posiciones:

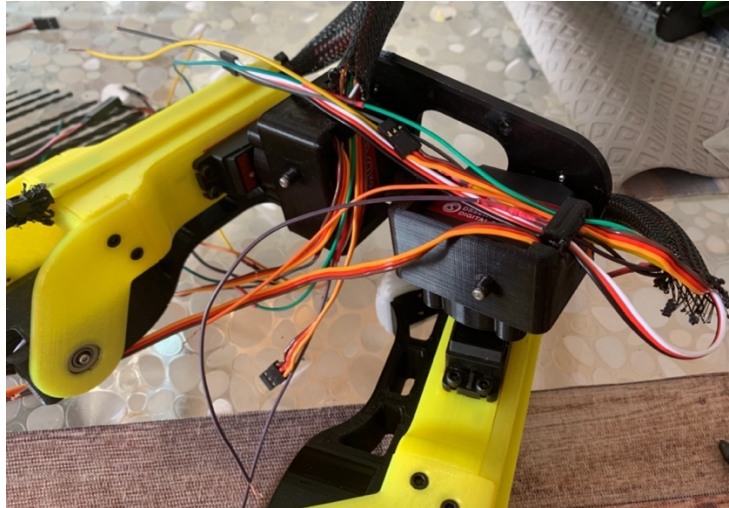


Figura 133. Piernas incorporadas a la montura del hombro f/e (Fuente propia).

- **Conexiones y cableado:** por último, se destacará la preparación de caminos de cableado (Figura 134) ya que no solo se tienen los cables propios de las piernas, sino que tanto la parte frontal como la trasera tienen un conjunto grande de cables que necesitan acceso al centro del robot. Estos huecos luego serán completamente ocupados por los mismos cables que lo facilitan conforme se van realizando labores de reparación. Conectado todo, podemos ver tanto el circuito (Figura 135) como el propio robot (Figura 136) en las siguientes imágenes:



Figura 134. Huecos para el cableado (Fuente propia).

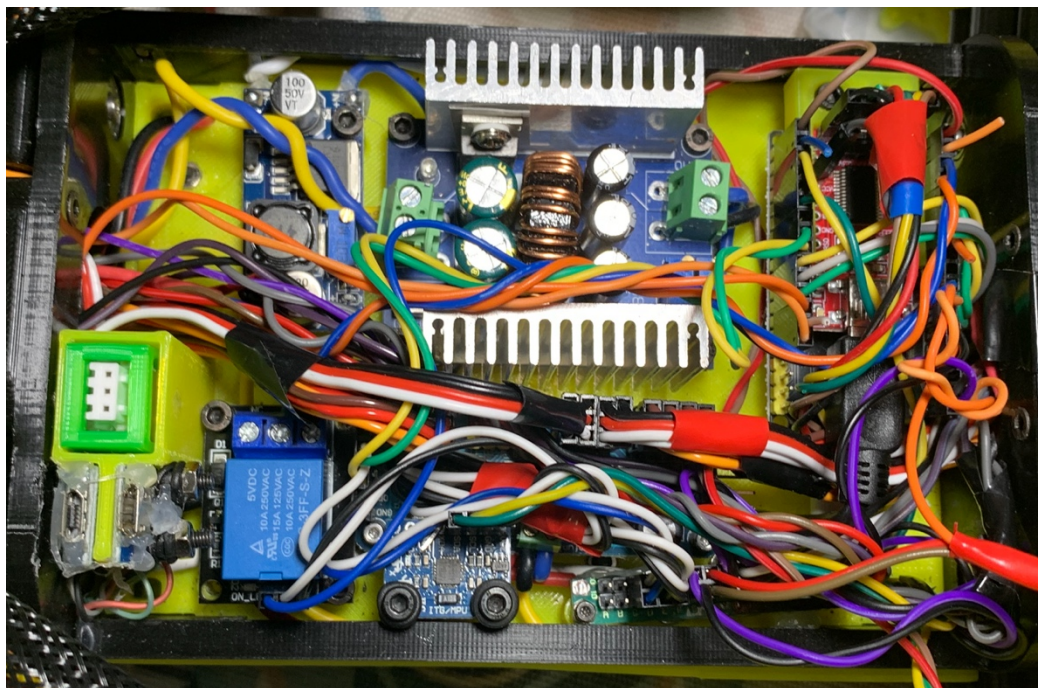


Figura 135. Cableado interno (v9\_final) (Fuente propia).



Figura 136. Robot terminado (v3.5) a escala con un Bichón Maltés (Fuente propia).



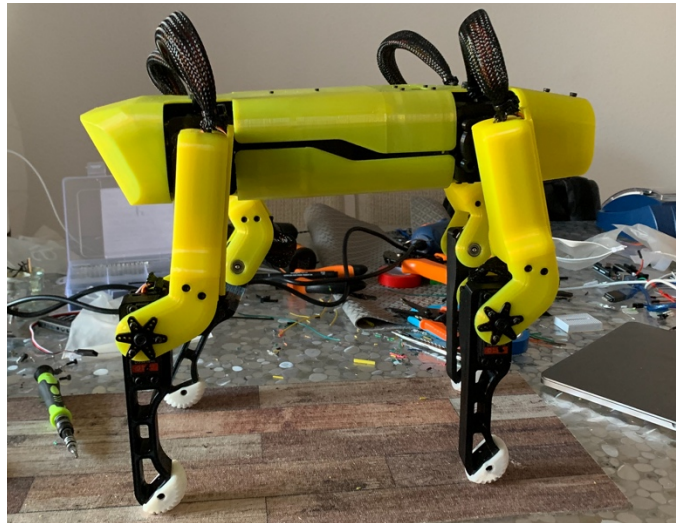


Figura 137. Robot terminado (v3.0) con las extremidades extendidas (Fuente propia).

#### 5.6.6 Construcción de una fuente de alimentación externa y cable de alimentación (adicional)

Finalmente, y antes de comenzar con la parte final de la fase 3, se decidió adquirir una fuente de alimentación externa para poder realizar pruebas más allá de la duración de la batería. Esto es especialmente interesante cuando se pretende verificar el aprendizaje de distintos agentes. Se presentaron 2 opciones: una fuente de PC convencional o una fuente como la de las impresoras 3D de 24V. La decisión fue la de adquirir una fuente de 500 W de PC que nos ofrece un gran abanico de salidas a distintas tensiones e intensidades. Sin embargo, para utilizarla primero es necesario identificar el cableado que normalmente se conecta a la placa de alimentación y que se encarga de conectar la fuente. Una vez localizado, se le instala un botón con una tapa impresa con la impresora como se puede ver en la Figura 138 a continuación:

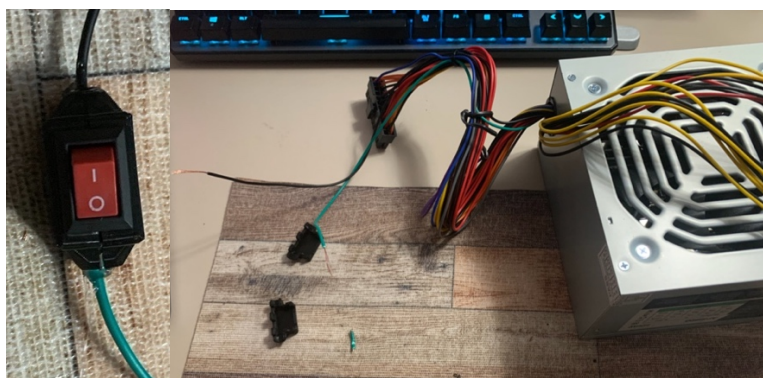


Figura 138. Elaboración del conector de la fuente de alimentación (Fuente propia).

Una vez localizados y unidos con el interruptor, se comprobó la polaridad y tensión de los distintos cables. Una vez localizado el mayor, 12 V a 25 A (amarillo) se le soldó un terminal macho de tipo T y a un cable eléctrico antiguo, se le soldaron 2 conectores macho teniendo en cuenta la polaridad a modo de alargador (Figura 139). Este conector pasaría por un hueco en la barriga del robot y se conectaría directamente a su entrada de alimentación de la batería la cual también emplea este tipo de conectores. Para una cuarta versión, se modificaría la tapa trasera para albergar un conector hembra con tapa que permitiese la conexión directa y mediante un relé, evitase el retorno de corriente a la batería.

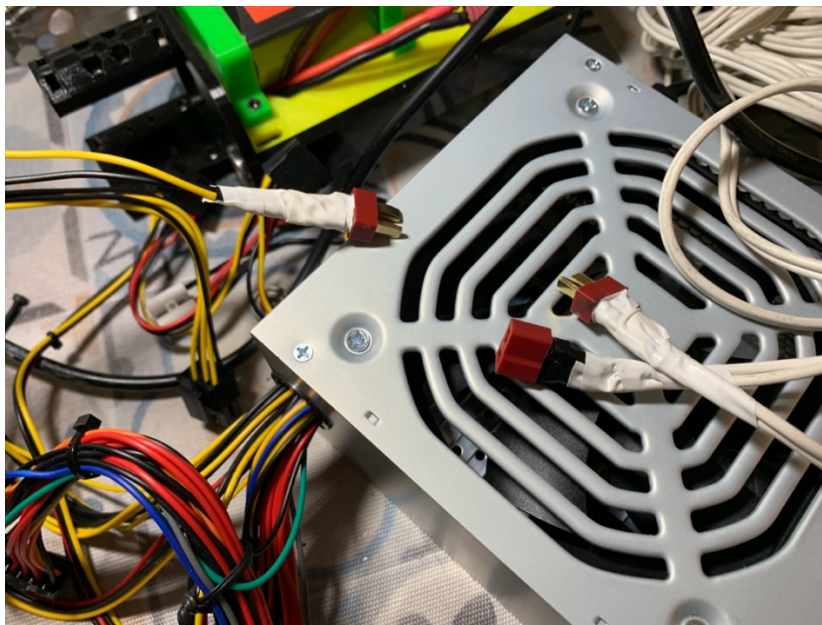


Figura 139. Elaboración del alargador de la fuente de alimentación (Fuente propia).

## 5.7 Control, programación y simulación: Fase 3

Con los sistemas verificados y el robot montado, se procedió a la última etapa de la fase 3 en la que se integraría todo el *software* con el *hardware*. El objetivo es generar un único programa de control para el robot y sus sensores sobre el que en una futura implementación o fase 4, implementar los comportamientos complejos que se pretenden preparar en simulación en esta fase.

### 5.7.1 Preparación del entorno de programación

Hasta el momento todos los códigos se han implementado en Arduino y por tanto, el sistema operativo en el que se realice la programación no era restrictivo. Llegados a este punto del desarrollo se decidió traspasar todos los archivos y librerías usadas a Linux pues se iba a incorporar *ROS* a la programación. Se consideró emplear un Macintosh para *ROS* pero se optó finalmente por Linux ya que contaba con un mejor equipo a nivel técnico.

La versión del sistema operativo recomendada para la implementación del paquete *Open Quadruped* [36] es Ubuntu 18.04 ya que emplea ROS Melodic. Por tanto, se realizó una instalación nueva con dicho sistema. Sin embargo al tratar de usar los paquetes, habían errores debidos a versiones y por tanto, se decidió ignorar el requisito de los desarrolladores y se instaló Ubuntu 20.04 LTS por lo que se trabajaría con ROS Noetic (no soportado).

Los pasos seguidos para la instalación fueron instalar todas las dependencias listadas, las librerías de Arduino y todos aquellos paquetes que no estaban listados como dependencias pero que eran necesarios. Las listas con esta información se hayan en la carpeta [compartida en la nube](#). Instalados todos estos paquetes y habiendo clonado la rama antigua de desarrollo (la de *spotmicroAI*), se creó e instaló el *catkin workspace*.

Por último se instalaron aplicaciones para grabar la pantalla, *IDEs* para programar y otras aplicaciones útiles para trabajar en el sistema Ubuntu.

### 5.7.2 Comprobación del funcionamiento de los paquetes instalados

Tras instalar una a una todas las librerías de Arduino e instalar la placa ESP32, se probó a compilar todos los códigos para verificar el correcto funcionamiento. Fue necesario instalar alguno adicional por problemas de compatibilidad, pero finalmente, se dispuso del entorno de programación que funcionaba mucho mejor que en *Mac OS* donde fallaba un 80% de las veces la conexión serie.

En el caso de la comprobación de los paquetes de *ROS* se hicieron las comprobaciones manuales para verificarlo. Saltaron una serie de errores relacionados con las versiones de ROS, pero si se pudo verificar la integridad de la instalación. Estos problemas deberían de ser solventados más adelante (punto 5.7.5) aunque eran esperables.

### 5.7.3 Elaboración del código de control básico, pruebas y correcciones del robot

Finalmente, tras 3 fases de desarrollo del diseño, llegó el momento de desarrollar el código de control básico del robot. El objetivo de este código es establecer una base para el controlador a bajo nivel. Para ello, el código debería de ser capaz de capturar y leer los datos de todos los sensores, tener un control sobre todos sus grados de libertad y poder manipular aquellos componentes de *feedback* al usuario. Por tanto, vista la implementación hasta el momento y los objetivos a alcanzar, se optó por una implementación modular y progresiva.

El primer paso consistió en integrar el **sistema visual** directos. Esto incluye la ESP32-CAM, los sensores de Ultrasonidos y el anillo *LED*. Nótese que no se incluye la pantalla de 0.96" *OLED* ya que esta funciona mediante el protocolo I2C y está sujeta al multiplexor.

- Inicialmente se implementó el mismo código para el anillo *LED* que en el de prueba, no obstante, esto bloqueaba el sistema en cada ejecución para la animación. Por ello, se cambió de tal forma que tras cada ciclo, aumentase en uno el número de luces conectadas. De este modo podríamos *debugear* la ejecución del código y demostrar el control sobre este componente.
- Para la cámara, bastó con cargar el ejemplo por defecto para el servidor de retransmisión inalámbrica. Hubo un pequeño problema con el cableado ya que como se mencionó en la fase 1, no se había podido soldar el cable RST de la placa al TTL y por tanto no funcionaba la conexión. Tras aplicar alcohol, rascar y eliminar la capa protectora, se pudo soldar un cable y funcionó.
- Por último, tenemos los sensores de ultrasonidos. Su código es muy sencillo y funciona de forma similar al sensor de presión en tanto a que nos devuelve un valor de tensión que procesaremos para obtener la distancia. Una vez cargado funcionó correctamente. Hay que tener en cuenta que ambos comparten TRIG pero podemos leerlos en 2 canales distintos de ECHO.

El segundo paso consistió en la lectura de los **sensores de alimentación**. Estos sensores son necesarios ya que nos brindan información relevante al estado de la carga de la batería (mediante la tensión) y de la cantidad de corriente que está pasando por el circuito para poder proteger el equipamiento.

Generalmente el aumento de esta es provocado por los servomotores y por tanto, en caso de peligro, es a estos a quienes se les corta la alimentación. Incluimos en este paquete al relé aunque sea un actuador.

- Para ambos sensores, se emplea directamente el código desarrollado en las pruebas. Una vez en ejecución es capaz de distinguir si está siendo alimentado por USB (~3.3 V), por la fuente interna (8.4-7.4 V) o por la externa (12 V). Para garantizar una lectura correcta, se toman 150 medidas y se computa el resultado.
- En el caso del relé, es como se comentó, muy simple ya que depende del estado de un bit. Debido a que tenemos la placa PCA conectada de forma normalmente abierta, deberemos activarlo para mover el robot.

El siguiente sensor en ser incorporado fue el de **presión**. El código era una lectura directa del pin analógico, pero fue aquí donde aparecieron el 80% de todos los problemas con el robot. La decisión tomada de emplear cables por tramos resultó ser demasiado endeble y se partían con el movimiento normal. El resultado era que había lecturas fantasmas de los sensores. Durante 1 semana y media entera se procedió a la reparación completa de cada pierna y su vuelta a la construcción. Resultó que el causante era la tensión a la que estaban sometidos en la entrada al cuerpo, ya que sólo en una ocasión partieron de forma accidental. La solución fue añadir nuevos segmentos que actuaran a modo de *buffer* o muelle y así evitar que los puntos de soldadura quebrasen. Hablando de estos, se aplicó una técnica de trenzado con más material que proporcionaba uniones más sólidas. Sin embargo, la solución real, hubiera sido reemplazarlos todos por cables enteros (sin tramos) desde el antebrazo al circuito principal. Tras la semana completa de reparaciones y haber reparado cada pierna 2 veces de media (debido a que cada vez se veía afectado uno o dos cables y suponía desmontarla por completo), se consiguieron lecturas limpias no afectadas por el movimiento y se pudo continuar con el desarrollo.

El siguiente paso natural era el control de los **servomotores**. Para ello partiríamos de la implementación de prueba y elaboraríamos una tabla de control. El objetivo era sencillo: lograr que se pudiese poner de pie y mantener su posición aún aplicándole presión. El programa, una vez aplicada la fórmula correcta para hallar los ángulos, funcionó a la perfección consiguiendo que el robot se levantara. Sin embargo, al poco perdía fuerza y se caía. Fue en este punto cuando volvió a salir humo del circuito, esta vez del módulo de control. La solución volvió a pasar por adquirir otra unidad y reemplazar la ESP32 que quedó dañada. Esto supuso un retraso de 5 días tras los cuales, se instalaron los

componentes, reguló el transformador para proporcionar toda la corriente posible (20 A) y funcionó (Figura 140). El robot era por fin capaz de ponerse de pie. Sin embargo, debido al factor del error humano, había una descompensación mínima en la montura de los servos que, tras corregirse al día siguiente, seguía presentando una pequeña variación de aproximadamente 5° que se solventó mediante *software*.

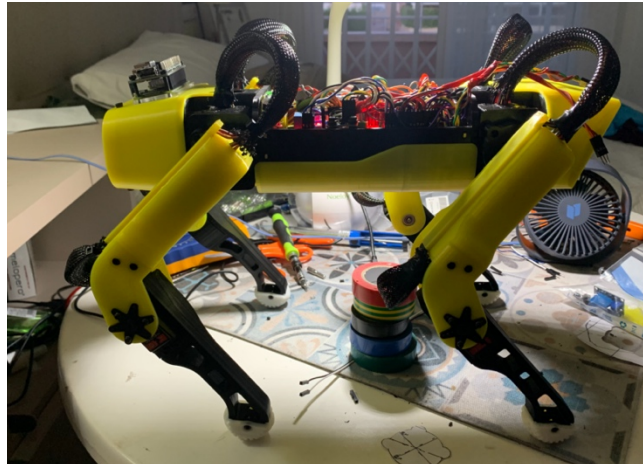


Figura 140. Primera posición estable (Fuente propia).

Con todo esto implementado, solo quedaba verificar el **multiplexor de I2C, la pantalla y los sensores inerciales**.

- El código de la pantalla se basaba en un hola mundo clásico al que se le sustituyó la frase por la información de la tensión y corriente. Sin embargo, resultó dañada y el repuesto llegaría hasta pasada la fecha del trabajo, sin embargo, se tenía redundancia con el anillo LED para indicar el estado de la batería si fuese necesario.
- Por suerte, ni la MUX ni los *IMUs* resultaron dañados y se pudo reutilizar el mismo código de prueba que se usó en el pasado con la salvedad de que al necesitar un segundo canal I2C adicional, habría que indicarle a la función *wire.begin()* los puertos SDA y SCL correspondientes. Hecho esto, se obtuvieron los datos y completamos el sistema.

#### 5.7.4 Evaluación del control básico

La mejor forma para visualizar esta parte es mediante los vídeos en la [carpeta compartida](#) no obstante, se proporcionarán algunas imágenes de las pruebas sobre las cuales se podrá generar una evaluación objetiva de los resultados en el próximo capítulo.



La primera imagen (Figura 141) muestra el robot apagado seguido de la Figura 142 en la que lo podemos ver unos instantes después en posición de apoyo completo:

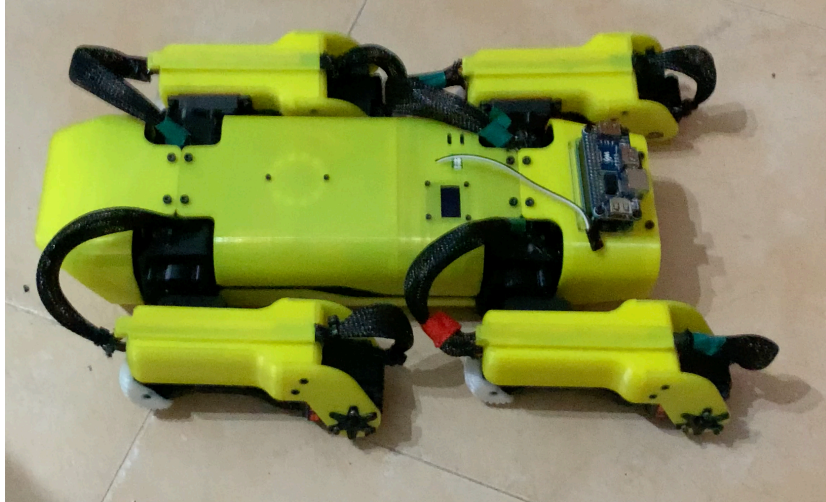


Figura 141. Robot apagado (Fuente propia).

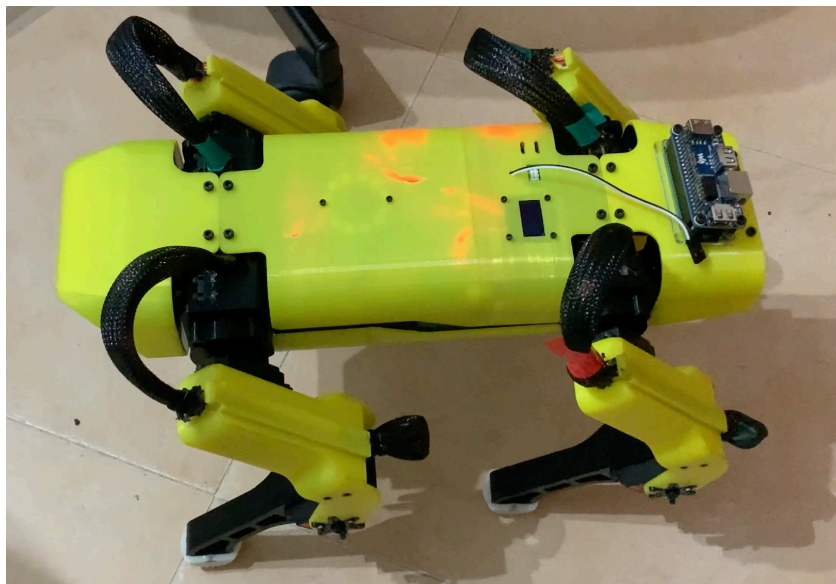


Figura 142. Robot en pie y en una posición estable (Fuente propia).

A continuación, en la Figura 143 se puede observar con más sistemas integrados durante una prueba de presión, en la Figura 144 se puede ver durante una prueba de los sensores de distancia frontales y finalmente en la Figura 145 podemos observarlo con el servidor web:

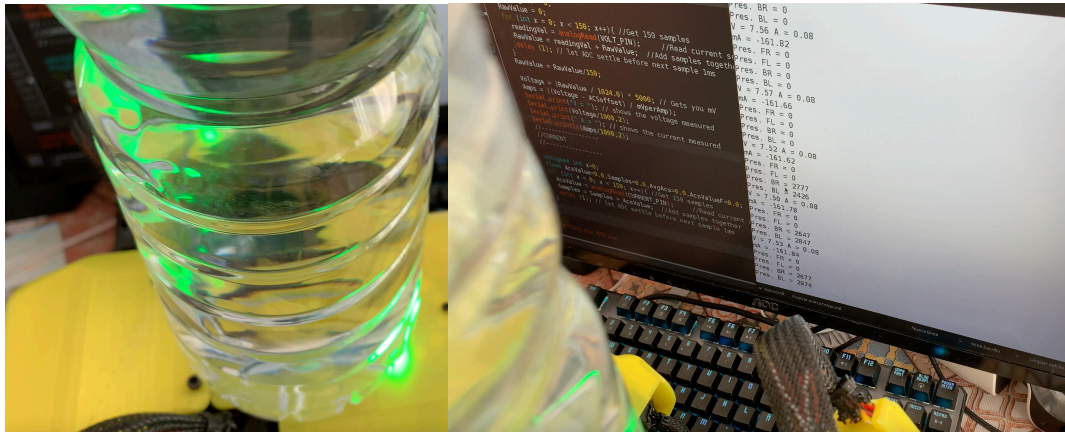


Figura 143. Prueba completa con sensores de presión y 1.5 kg de agua (Fuente propia).

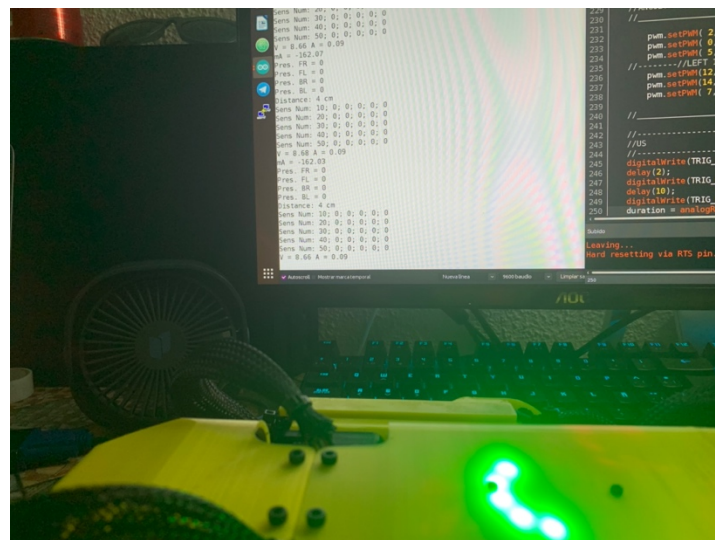


Figura 144. Prueba de los sensores de distancia (Fuente propia).



Figura 145. Prueba del servidor web con imagen en directo (Fuente propia).

Por último, tenemos los sensores Inerciales. Para probarlos se movió el robot con los servomotores en posición neutra a distintas inclinaciones y alturas y se grabó la salida por el puerto serie. Como se puede ver en las figuras 146 a 148, son registrados por los *IMUs* del lateral derecho, izquierdo y central:

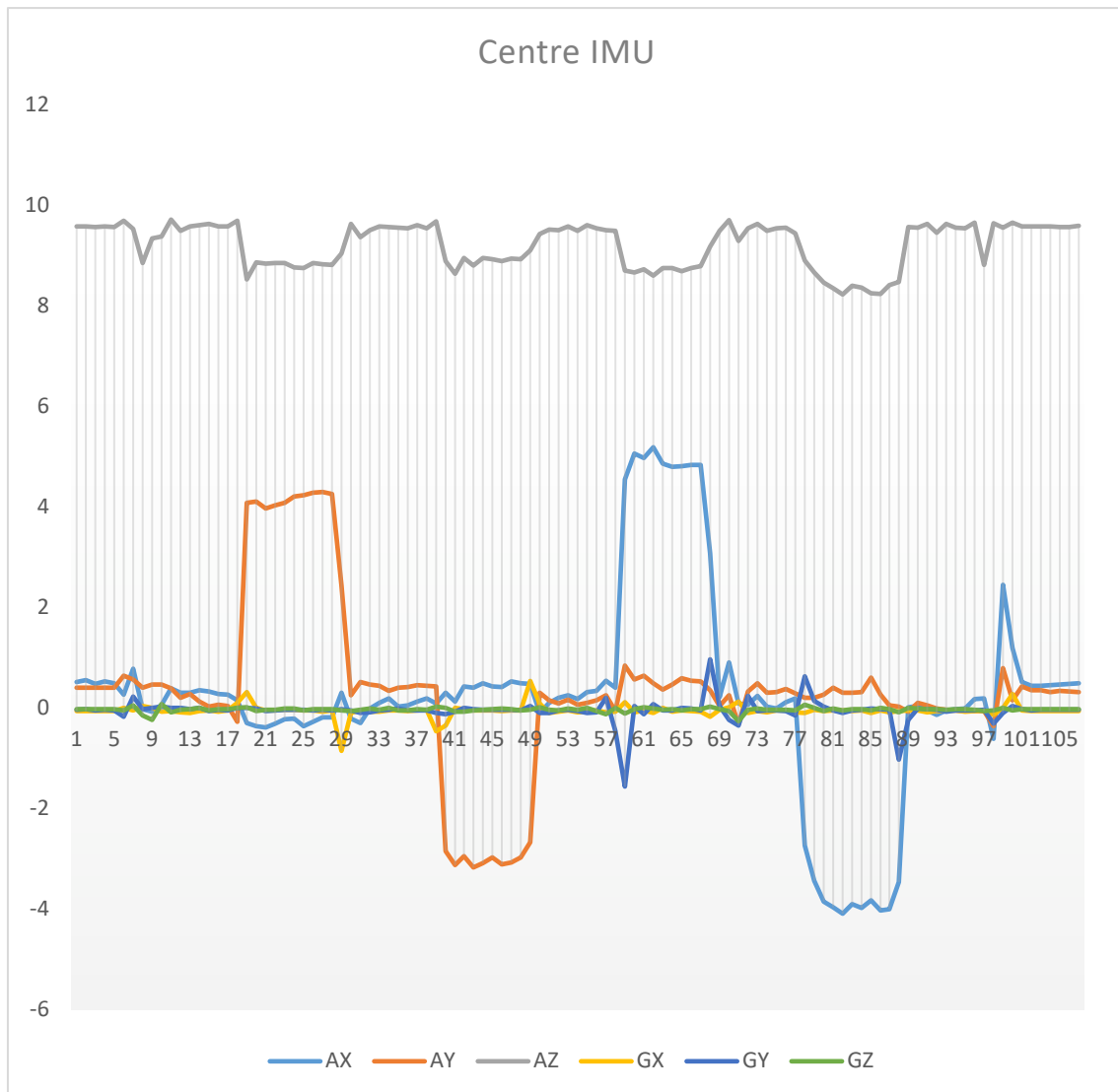


Figura 146. Lecturas del IMU central (Fuente propia).

Para las medidas correspondientes a los laterales, se mostrará una de las dos ya que la variación es mínima debido a que mantienen la misma posición y lo único que podría variar es el ángulo en el movimiento. El objetivo es mostrar la captura y la posibilidad de su usabilidad posterior.

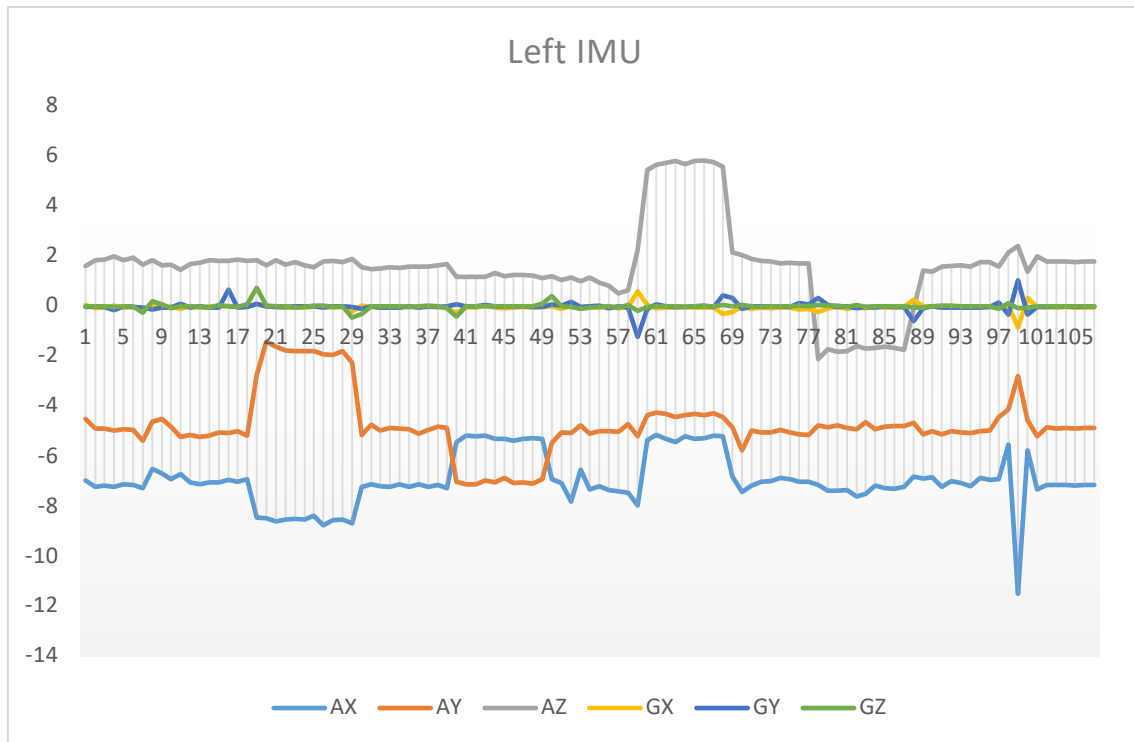


Figura 147. Lecturas del IMU izquierdo (Fuente propia).

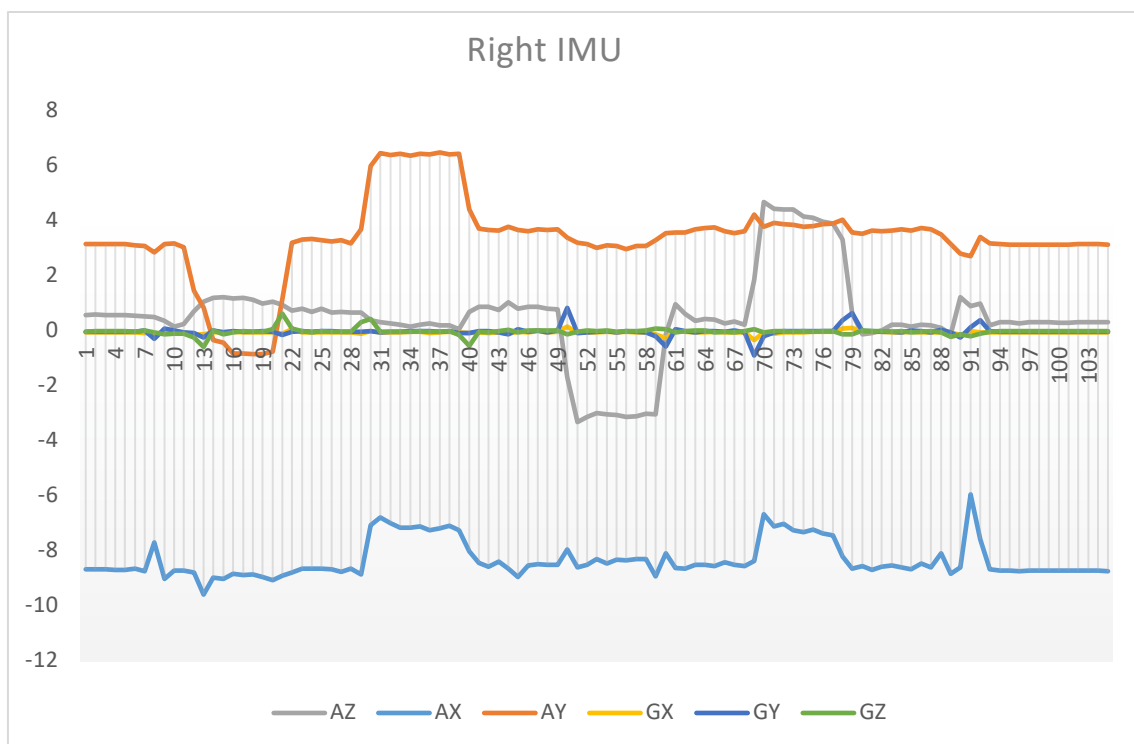


Figura 148. Lecturas del IMU izquierdo (Fuente propia).

### 5.7.5 Adaptación de los paquetes y códigos base a ROS Noetic y al prototipo

Una vez conseguido el código de control básico, llegó el momento de desarrollar la simulación con *RL* con la que, en trabajos futuros, controlar al robot. Los pasos que se deberían seguir para lograr esto fueron:

- Instalar los paquetes
- Resolver problemas de dependencias e incompatibilidades
- Arreglar códigos antiguos
- Actualizar el modelo URDF (numéricamente)
- Verificar la usabilidad del sistema

El primer punto se completó en el 5.7.1 pero detectamos distintos errores. Estos, tras analizarlos, tienen su origen en un problema de Melodic a Noetic que resulta en la imposibilidad de localizar la distribución correcta de Python en el ordenador. La solución consistió en reinstalar Python 3 eliminando la distribución de Python 2 previamente y manualmente indicando un comando a cada código para que supiese localizar las librerías: `#!/usr/bin/env Python`.

A raíz de esta solución se tuvieron que reinstalar algunos paquetes adicionales para *ROS-Control* entre otros y drivers para dispositivos joysticks. Se modificó el código encargado de gestionar el movimiento del robot mediante el mando (enlazado a *spot\_move.launch*) ya que no implementaba el uso de agentes para el movimiento y por tanto solo usaba un lazo abierto de control.

Finalizando las modificaciones, se cambió el modelo URDF a nivel matemático para que coincidiesen las cinemáticas y se comprobó que el código encargado de la cinemática inversa aplicase esta de forma correcta. No se ha cambiado el modelo 3D pues los cambios son meramente estéticos y las discrepancias de medidas son visualmente indistinguibles y solo atañen al hombro abductor. Tras estas correcciones y cambios, se lanzaron las distintas aplicaciones con retornos positivos.

### 5.7.6 Verificación de la comunicación con ROS y mando remoto

Partiendo de un entorno ya verificado, se probó a conectar un joystick para así evaluar los cambios realizados sobre el modelo. Sin embargo y pese a ser compatible, contar con los drivers y dar lecturas correctas por parte del nodo del



joystick, nos se pudieron implementar ni el *Dualshock 4* (Figura 149a) ni el 3 de Sony. La solución fue adquirir el controlador Logitech Gamepad F310.



Figura 149a. Lecturas del DS4 en el nodo de ROS (Fuente propia).

Al conectar el nuevo mando, si se pudo controlar el robot (Figura 149b) como se puede ver en uno de los [vídeos compartidos](#).

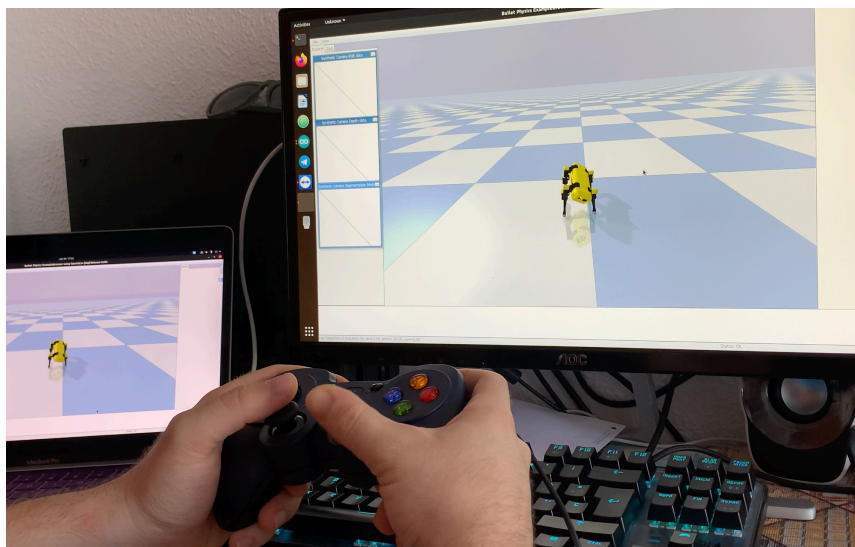


Figura 149b. Ejecución con el gamepad F310 de *spot\_move* (Fuente propia).

Para ejecutar el proceso visto en la Figura 149, es necesario seguir estos pasos:

1. Iniciar roscore
2. Abrir el espacio de trabajo: `cd catkin_ws`
3. Hacer un source `devel/Setup.bash` para disponer de los paquetes
4. Obtener el ID del mando con: `ls/dev/input/` (saldrá jsNumero)
5. Poner a disposición de ROS el mando: `sudo chmod a+rw/dev/input/jsNº`
6. Ejecutar: `roslaunch mini_ros spot_move.launch`

Una vez dentro (Figura 149) podremos mover al robot con los siguientes comandos mientras realizamos un control con el agente 149 entrenado en el control en lazo abierto o sin él:

- **Botones generales:**

- A: cambiar entre paso a paso y RPY
- X: E-STOP (activar y desactivar)
- Gatillo superior derecho / izquierdo: restablecer todos los valores predeterminados

- **Modo paso a paso:**

- Joystick derecho arriba / abajo: longitud del paso
- Joystick derecho izquierdo / derecho: fracción lateral
- Joystick izquierdo arriba / abajo: altura del robot
- Joystick izquierdo Izquierda / Derecha: Velocidad de guiñada
- Teclas de flechas arriba / abajo (DISCRETO): altura del paso
- Teclas de flechas izquierda / derecha (DISCRETO): profundidad de paso
- Gatillo inferior derecho / izquierdo: velocidad de paso (modular)

- **Modo de visualización:**

- Joystick derecho arriba / abajo: Cabeceo
- Joystick derecho Izquierda / Derecha: alabeo
- Joystick izquierdo arriba / abajo: altura del robot
- Joystick izquierdo Izquierda / Derecha: guiñada

Verificado que es posible el control del robot, la cinemática inversa y que efectivamente, la simulación funciona como es esperado, se procedió al punto final en el que se trataría de realizar el entrenamiento en distintas condiciones y evaluar cuan de viable sería su futura implementación en la realidad.

### 5.7.7 Estudio del algoritmo GMBC y su modelo de aprendizaje por refuerzo

Antes de proceder con el entrenamiento del modelo y la evaluación, es necesario comprender cómo funciona el algoritmo que tiene detrás para poder después evaluar correctamente los resultados obtenidos.

En el paper *Linear Policies are Sufficient to Enable Low-Cost Quadrupedal Robots to Traverse Rough Terrain*[36], los investigadores plantean, como bien dice el título, que una política lineal es suficiente para modular un generador de trayectorias en lazo abierto. Esto últimamente implica que un cuadrúpedo pueda caminar sobre un terreno desconocido y con relieves, sin mayor dificultad ni programación.

El control de la marcha se basa en el algoritmo visto en la [Figura 31](#) en cuanto a que el controlador trata de generar una curva de movimiento óptima que permita un espacio libre de obstáculos y mantenga la estabilidad en el movimiento (Figura 150). Esto es básicamente una curva de Bézier. El inconveniente, como ya se ha mencionado es que la plataforma carece de los sensores apropiados para su uso correcto sin embargo, con pocos datos y aprendizaje por refuerzo, si se podría lograr un avance significativo.

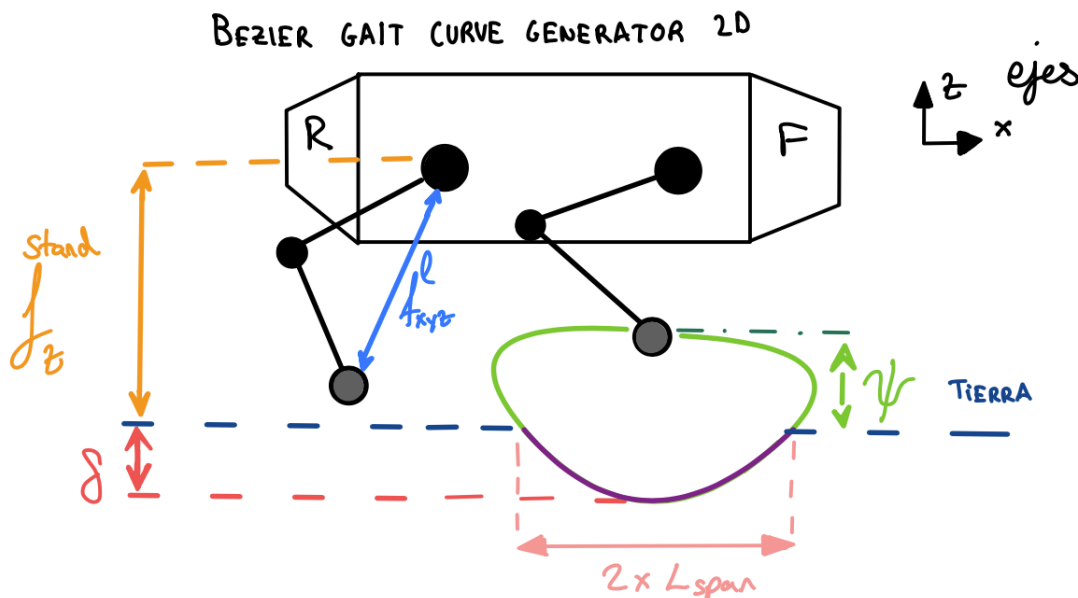


Figura 150. Modelo de del posicionamiento del pie del robot en las curvas de Bézier (Fuente propia).



Visto desde arriba (Figura 151) podemos observar como siempre se trata de mantener la curva tangente a la distancia mínima al origen del movimiento por lo que aspira a un futuro posicionamiento estable siempre dentro de una zona de actuación y seguridad. En el caso de trasladarnos a las 3 dimensiones (Figura 152) observamos como se orientan de forma opuesta para conseguir el desplazamiento óptimo en esa posición:

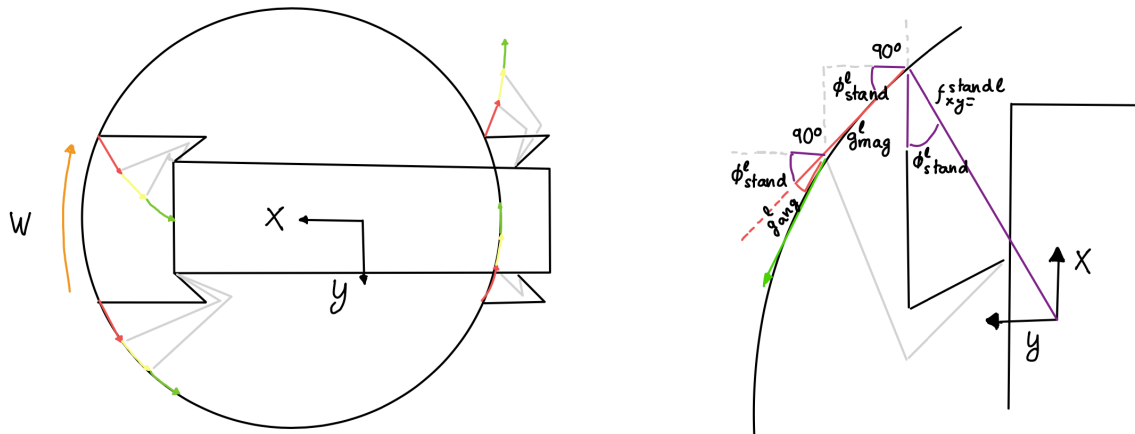


Figura 151. Esquemática de del posicionamiento del pie del robot en las curvas de Bézier (Fuente propia).

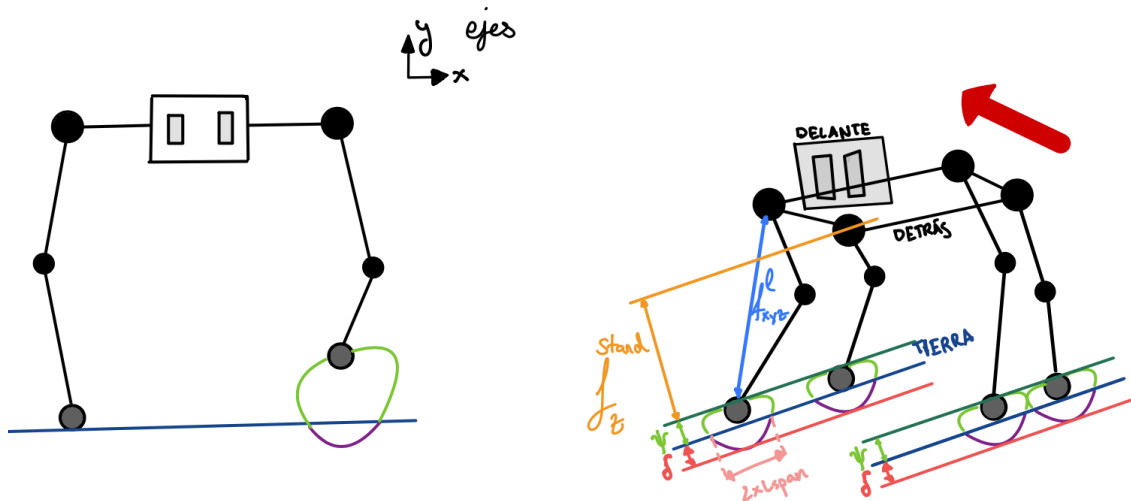


Figura 152. Esquemática de del posicionamiento del pie del robot en las curvas de Bézier en movimiento hacia delante (Fuente propia).

Por tanto, el agente, tratará de proporcionar la mejor curva para garantizar el máximo agarre, estabilidad y agilidad. Sin embargo, lo que dicen haber desarrollado es un sistema de lazo abierto con política lineal y con un único sensor inercial que aplica estas curvas, por lo que tiene gran potencial. El siguiente paso es entender los dos últimos algoritmos presentados: El algoritmo de aprendizaje automático (Figura 153) y el generador de curvas por pierna (Figura 154):

---

**Algoritmo RL:** Simulación RL para uso de DR-GMBC usando la búsqueda aleatoria aumentada.

---

*Se inicializa:* una política con parámetros  $\theta_0$ , dominio de distribución  $\mathbf{P}$ , función de recompensa  $\mathbf{R}$ , **GMBC** (Algoritmo 1), número de iteración  $k=0$ , construye **ARS**.

1. Mientras que el entrenamiento no converja haremos:
    - $\sigma \sim \mathbf{P}$  muestreo de los parámetros del dominio
    - **ARS** a paso de (1) con **randomización del dominio + GMBC**
    - $\theta_{k+1} \leftarrow \mathbf{ARS}(\pi, \theta_k, \mathbf{R}, \sigma, k)$
    - $k \leftarrow k + 1$
  2. Finalizamos el bucle
    - **Return**  $\theta_k$
- 

Figura 153. Algoritmo de aprendizaje automático por refuerzo (Fuente propia).

---

**Algoritmo generador:** Generador curva Bezier  $\Gamma$  - Por pierna  $l$ .

---

*Dada una entrada:*  $\zeta$ .

1. Mapea  $t$  a la fase del pie  $S_l(t)$  usando  $\rightarrow$
2.  $(f^{tr}_q, f^{tr}_z) = \gamma(S_l(t))$  usando  $\rightarrow$
3.  $(f^{yaw}_q, f^{yaw}_z)$
4.  $(f^{tr}_{xyz}) = [f^{tr}_q \cos p \quad f^{tr}_q \sin p \quad f^{tr}_z]$
5.  $\Phi^{l}_{arc} = g^{l}_{ang} + \Phi^{l}_{stand} + \pi/2$
6.  $f^{yaw}_{xyz,l} = [f^{yaw}_q \cos \Phi^{l}_{arc} \quad f^{yaw}_q \sin \Phi^{l}_{arc} \quad f^{yaw}_z]$
7.  $f^{l}_{xyz} = f^{tr}_{xyz} + f^{yaw}_{xyz,l} + f^{stand}_{xyz}$
8. Devolvemos  $f^{l}_{xyz}$  al robot para la actuación sobre los accionamientos

$$S_l(t) = \begin{cases} \frac{t_\ell}{T_{stance}} & 0 < t_\ell < T_{stance} \\ \frac{t_\ell + T_{stride}}{T_{stance}} & -T_{stride} < t_\ell < -T_{swing} \\ \frac{t_\ell + T_{swing}}{T_{stance}} & -T_{swing} < t_\ell < 0 \\ \frac{t_\ell - T_{stance}}{T_{swing}} & T_{stance} < t_\ell < T_{stride} \end{cases}$$

Figura 154. Algoritmo generador de curvas para cada pierna (Fuente propia).

En el primer caso, se trata de un algoritmo de aprendizaje por refuerzo con política lineal con la salvedad de que hay una randomización del dominio para el aprendizaje (se varían datos del robot) y se aplica la curva generada al espacio de acciones. En el caso del segundo, se puede ver como el algoritmo efectivamente se basa en la lectura de un sensor inercial y geometría para trazar

el siguiente punto de la trayectoria. A continuación, se puede ver una ilustración (figuras 155 y 156) de como realiza los cálculos:

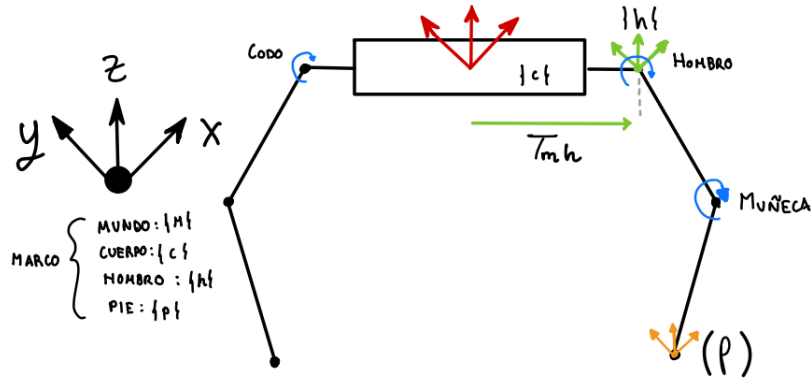


Figura 155. Posición neutral del robot (Fuente propia).

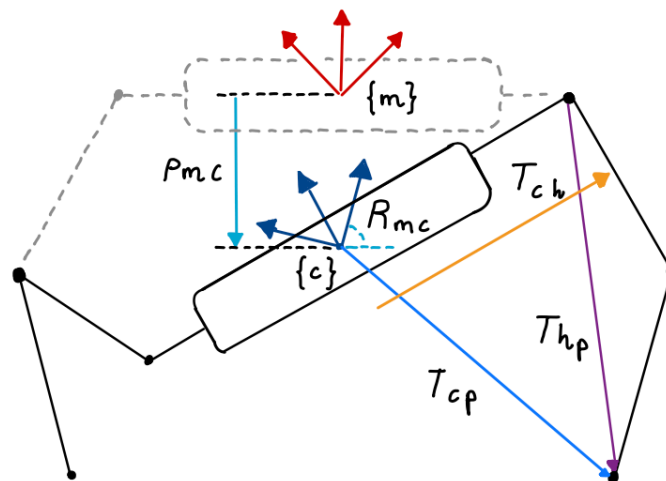


Figura 156. Cinemática inversa a partir de la variación del ángulo y la aceleración del sensor inercial y la propia cinemática del robot (Fuente propia).

Esta implementación es similar a la que se pretende implementar en un futuro con el sistema *IMU-ENCODER* ideado para el prototipo. Por tanto, con unas futuras modificaciones, al modelo, red y entorno, se podría enriquecer el aprendizaje con el abanico de información proporcionada por los sensores equipados en el modelo de la fase 3.

Comprendida la implementación (más detallada en su artículo), se procedió al último paso de verificación y entrenamiento para finalizar la fase 3.

## 5.7.8 Entrenamiento del modelo de marcha sobre terreno irregular con aprendizaje por refuerzo y el algoritmo GMBC

En primer lugar, se deberá preparar de nuevo el entorno del sistema para ejecutar los programas. En este caso particular, para ejecutar el aprendizaje automático deberemos de realizar los siguientes pasos:

1. Iniciar roscore
2. Abrir el espacio de trabajo: `cd catkin_ws`
3. Hacer un source `devel/Setup.bash` para disponer de los paquetes
4. Navegar hasta la carpeta: `src/spot_bullet/src`
5. Ejecutar: `python3 spot_ars.py (-hf si queremos que entrene con terreno aleatorio)`
6. Ya estaríamos entrenando como en la Figura 157
7. Cerrar la ventana cuando se desee parar

```
jon@jon-noetic:~/catkin_ws/src/spot_bullet/src$ python3 spot_ars.py -hf
pybullet build time: Jun 10 2021 19:56:47
/home/jon/catkin_ws/src/spot_bullet/src/../../spotmicro/util/pybullet_data
STARTING SPOT TRAINING ENV
argv[0]=
argv[0]=
b3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
No inertial data for link, using mass=1, localInertiaDiagonal = 1,1,1, identity local inertial frameb3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
front_left_leg_link_coverb3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
No inertial data for link, using mass=1, localInertiaDiagonal = 1,1,1, identity local inertial frameb3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
front_right_leg_link_coverb3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
No inertial data for link, using mass=1, localInertiaDiagonal = 1,1,1, identity local inertial frameb3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
rear_left_leg_link_coverb3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
No inertial data for link, using mass=1, localInertiaDiagonal = 1,1,1, identity local inertial frameb3Warning[examples/Importers/ImportURDFDemo/BulletUrdImporter.cpp,126]:
rear_right_leg_link_cover[6, 11, 10, 21]
/home/jon/.local/lib/python3.8/site-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
TERRAIN SHAPE: 0
TERRAIN SHAPE: 1
Action SPACE: Box(-1.0, 1.0, (15,), float32)
STATE DIM: 16
ACTION DIM: 15
STARTED SPOT TRAINING ENV
Total T: 369 Episode Num: 0 Episode T: 368 Reward: -269.89 REWARD PER STEP: -0.73
Total T: 1308 Episode Num: 1 Episode T: 939 Reward: -184.80 REWARD PER STEP: -0.20
Total T: 1711 Episode Num: 2 Episode T: 483 Reward: -372.81 REWARD PER STEP: -0.93
Total T: 2085 Episode Num: 3 Episode T: 374 Reward: -452.88 REWARD PER STEP: -1.21
Total T: 2929 Episode Num: 4 Episode T: 844 Reward: -586.65 REWARD PER STEP: -0.70
Total T: 3413 Episode Num: 5 Episode T: 484 Reward: -321.84 REWARD PER STEP: -0.66
Total T: 4225 Episode Num: 6 Episode T: 812 Reward: -1074.09 REWARD PER STEP: -1.32
Total T: 5977 Episode Num: 7 Episode T: 1752 Reward: 1689.49 REWARD PER STEP: 0.96
Total T: 7667 Episode Num: 8 Episode T: 1690 Reward: 1394.76 REWARD PER STEP: 0.83
Total T: 9100 Episode Num: 9 Episode T: 1493 Reward: 1381.30 REWARD PER STEP: 0.93
Total T: 9935 Episode Num: 10 Episode T: 775 Reward: -17.32 REWARD PER STEP: -0.02
Total T: 10041 Episode Num: 11 Episode T: 106 Reward: -253.50 REWARD PER STEP: -2.39
Total T: 11197 Episode Num: 12 Episode T: 1156 Reward: 640.22 REWARD PER STEP: 0.55
Total T: 11474 Episode Num: 13 Episode T: 277 Reward: -259.99 REWARD PER STEP: -0.94
Total T: 12113 Episode Num: 14 Episode T: 639 Reward: -176.48 REWARD PER STEP: -0.28
Total T: 13192 Episode Num: 15 Episode T: 1079 Reward: -0.60 REWARD PER STEP: -0.00
Total T: 13553 Episode Num: 16 Episode T: 361 Reward: -277.62 REWARD PER STEP: -0.77
Total T: 13737 Episode Num: 17 Episode T: 184 Reward: -277.80 REWARD PER STEP: -1.51
```

Figura 157. Entrenamiento de la red por aprendizaje por refuerzo (Fuente propia).

Un par de horas después, tendremos una gran cantidad de agentes disponibles que podremos evaluar. Para ello simplemente tendremos que lanzar: `python3 spot_ars_eval.py` seguido de cualquiera de estos argumentos:

- h, --help mostrar este mensaje de ayuda y salir
- hf, --HeightField → utiliza el mapa de alturas en el entorno
- r, --DebugRack → coloca el robot fijo en el aire
- p, --DebugPath → Dibujar la ruta del pie de robot
- gui, --GUI → Permite controlar el robot con una GUI y observar reacciones a comportamientos inesperados

-a, --AgentNum → Número de agente a cargar (seguido de número) (a mayor más agresiva su política)

En la [carpeta compartida](#) se puede encontrar un vídeo probando estas dos últimas en profundidad. A continuación, se puede ver el resultado de cargar un agente aleatorio con el mapa de alturas (Figura 158) para evaluar si el agente es capaz de navegar por el terreno tras unos pocos episodios simulando e incluso modificando las condiciones del robot de forma manual (Figura 159):

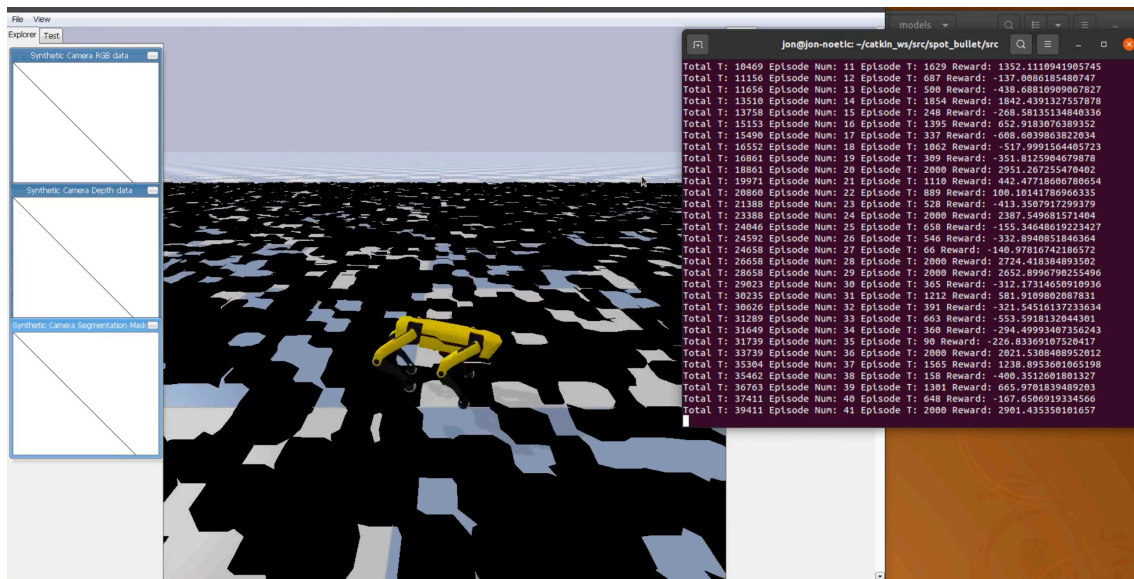


Figura 158. Evaluación de la red por aprendizaje por refuerzo (Fuente propia).

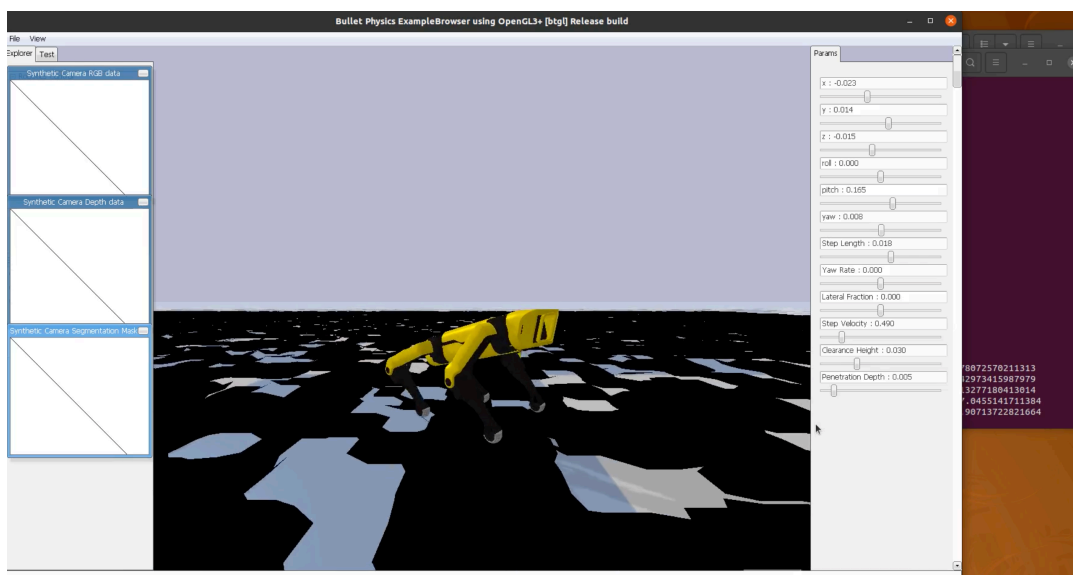


Figura 159. Evaluación manual de la red por aprendizaje por refuerzo (Fuente propia).

Obtenidos todos los datos se distribuyeron en tres gráficas para observar y evaluar objetivamente como había transcurrido el aprendizaje (Figura 160), la variación en la política de salida (figuras 161 y 162) y la supervivencia del agente comparando los dos métodos disponibles (figuras 163 y 164). De esta forma, se puede valorar si esta implementación puede llegar a funcionar correctamente en la plataforma robótica desarrollada en el futuro. Las imágenes se pueden ver a continuación:

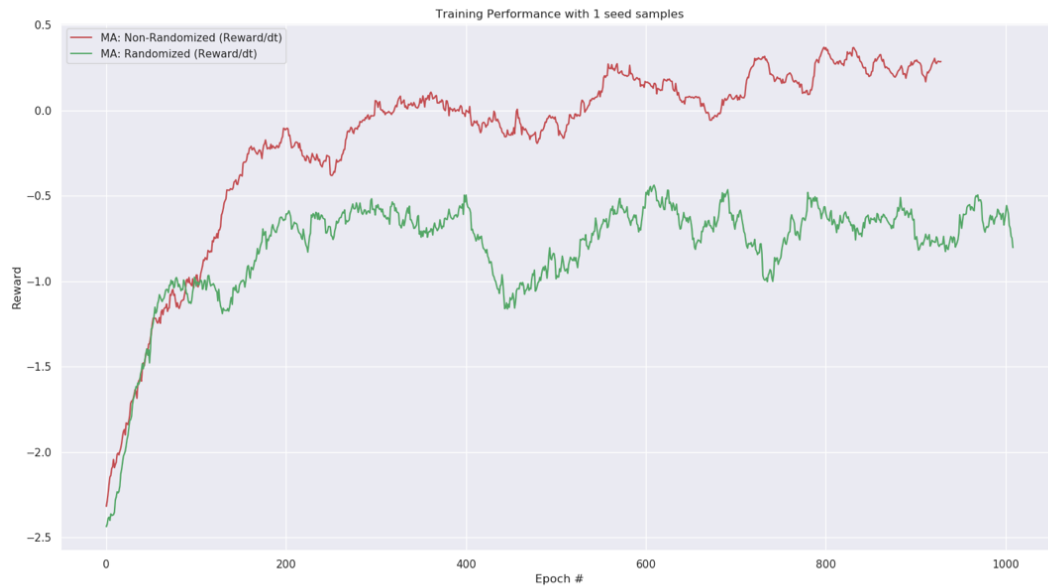


Figura 160. Comparación del rendimiento en el aprendizaje comparando el modelo no aleatorizado (rojo) frente al aleatorizado (verde) (Fuente propia).

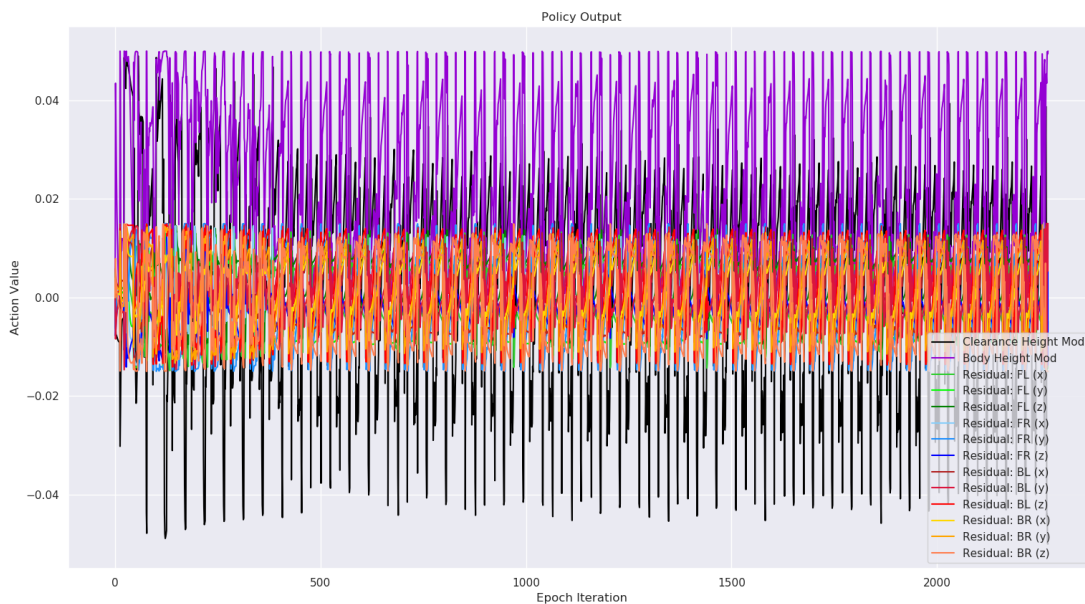


Figura 161. Política de salida para cada articulación (Fuente propia).

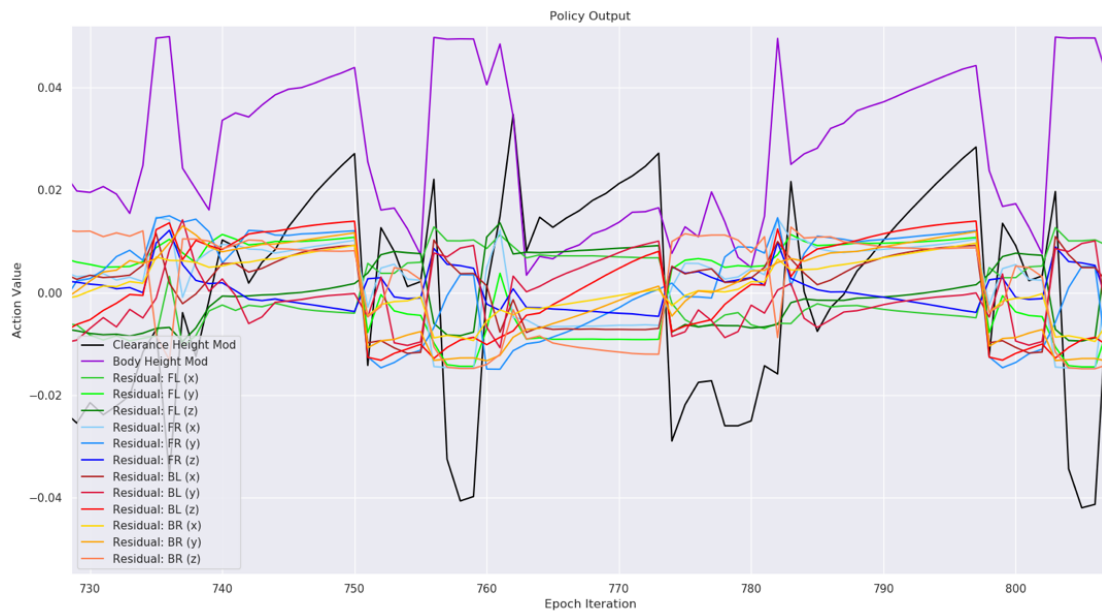


Figura 161. Política de salida para cada articulación (con zoom) (Fuente propia).

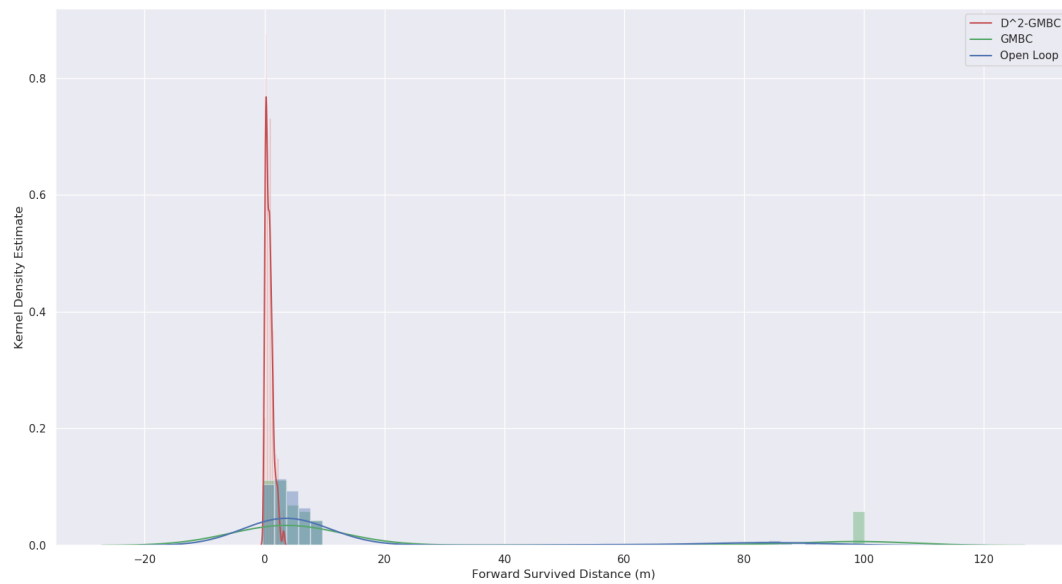


Figura 162. Comparativa de supervivencia entre métodos (zoom) (Fuente propia).

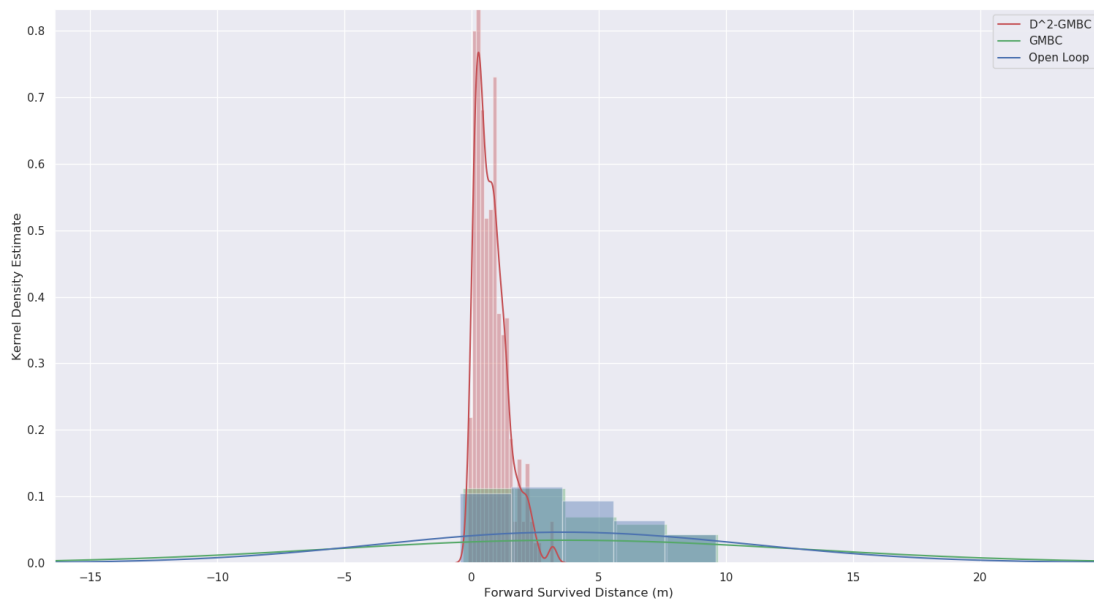


Figura 163. Comparativa de supervivencia entre métodos (sin zoom) (Fuente propia).

Obtenidos todos los datos y cumplidos los objetivos, se preinstaló una RPi ZW en el robot en la parte trasera y se le dio alimentación para dejarla preparada para una futura fase 4. Con esto último se da por concluido el desarrollo del trabajo final de Máster y se procede a evaluar el trabajo producido y los resultados aquí obtenidos en las conclusiones del capítulo sexto.



---

## Capítulo 6: Conclusiones

Tras haber realizado el desarrollo del proyecto se ha podido comprobar como el desarrollo en fases, aunque pueda parecer más costoso inicialmente, permite mejorar el sistema y aprender a sacarle el máximo partido de forma natural y progresiva logrando cumplir con todos los objetivos. Durante el transcurso de este aprendizaje se han adquirido conocimientos y competencias que de forma indirecta que han resultado ser bastante beneficiosos a lo largo del trabajo. También se han detectado una serie de puntos que podrían haber sido subsanados de haberse tomado otras decisiones como también puntos interesantes que presentan gran potencial.

En primer lugar, en cuanto a los defectos, hay que remarcar la elección de cable modular al haber supuesto una gran cantidad de horas en reparación, un 7 % de los 8 meses de desarrollo. Esto es corregible para futuras implementaciones tal y como se comentaba en el capítulo 5. La solución consiste en emplear cableados directos sin uniones y con zonas de respiro o *buffers* que puedan tolerar las tensiones producidas durante el movimiento.

En segundo lugar y a pesar de no estar directamente relacionado con la construcción o el diseño, los fallos de los dispositivos electrónicos, estos han supuesto un 9% del tiempo de desarrollo y no han podido ser predecibles. A pesar de haber probado todos los componentes por separado y de forma individual, se podría haber hecho un *dry run* de los componentes críticos para evitar el daño accidental sobre los secundarios.

En cuanto al desarrollo, hay que mencionar que se han podido cumplir todos los objetivos planteados e incluso sentado las bases para futuras implementaciones. Concretamente fase a fase, en la primera se consiguió aprender las técnicas de impresión 3D y construcción básica, la programación y usabilidad de los componentes, el diseño de circuitería electrónica y el análisis crítico de un modelo robótico existente. En la segunda fase se pudo aplicar los conocimientos adquiridos en el máster para el diseño de nuevos componentes, la mejora del sistema actual para cumplir con los requisitos de una tarea, ampliación de sistemas y gestión del proyecto. Por último, en la tercera fase se aplicó un rediseño y sensorización completa requiriendo técnicas de modelado, cambios en el diseño, adaptabilidad y soluciones limitadas al entorno. También fue cuando se introdujo el primer código general de control y se profundizó en una de las técnicas de estado del arte de control.

Dicha técnica basada en el aprendizaje por refuerzo con políticas lineales, resultó ser todo un éxito tras ser adaptada al robot actual en el capítulo 5.7. Observando las figuras con los resultados (160 a 163) podemos verificar el potencial que presenta en conjunto con la técnica de simulación de encoders con sensores inerciales propuesta en la fase 3. Esto es debido a que, con muy pocos datos de entrada y la cinemática del robot, es capaz de generar un control complejo en entornos desconocidos. Sin embargo, no solo se verificó el funcionamiento, sino que se adaptaron los códigos y programas a la última versión de ROS, se corrigieron problemas, se entrenó y se evaluó a dichos agentes sobre los que se han obtenido los datos. Esto sienta las bases para la hipotética cuarta fase de desarrollo del trabajo, en la que se integra con el robot actual utilizando la *RPI* instalada en la parte trasera.

En conclusión, se han logrado los objetivos iniciales de escoger un robot, analizar el estado de desarrollo actual de plataformas similares, simular y rediseñar, no solo la parte interna, sino finalmente el 85% del robot incluyendo un sistema de detección, el *USDAR* adicional junto con una expansión total de sistemas electrónicos. Sobre eso se ha aplicado una mejora progresiva, estudiada y detallada de cada iteración, se han solventado los incidentes que han ido surgiendo, se ha desarrollado un código de control básico total y desplegado completamente en el prototipo final. Finalmente se ha establecido una simulación y verificado la viabilidad de una novedosa técnica de control con aprendizaje por refuerzo que podría permitir dotar en la siguiente implementación de autonomía al robot.

Personalmente estoy muy satisfecho con el trabajo realizado pues he podido aplicar y ver como todos los conocimientos vistos en el máster han dado sus frutos en el prototipo final. Al comienzo y sobre todo al final, con los problemas técnicos, ha presentado batalla, pero el resultado ha merecido la pena no solo por el robot, sino por todo el conocimiento y destrezas que he adquirido para llegar a estas líneas. Sin duda seguiré trabajando en él y puedo decir que ya he empezado con la fase 4 con muchas ganas e ilusión de ver hasta dónde puede llegar.

Por último, en la décima y última tabla, se pueden ver todos los costes relacionados con la ejecución del proyecto a lo largo de estos 8 meses de desarrollo:

COSTES TOTALES DEL DESARROLLO				
Nombre	Modelo	Cantidad	Precio €	Total
Servomotores	MG996R Servo 12Kg	12	5	60
Microcontrolador	ESP32-DevKitC	1	9	9
Cámara	ESP32-CAM	1	10	10
Conexión TTL	FT232RL USB-TTL-Adapter	1	8	8
Motores	DS3218MG Servo 20kg	12	11	132
Protector de cables	3m de malla de cable	1	8	8
Protector de cables	Tubos termoretráctiles	1	15	15
Extensor de cables	Servocable Extension	4	10	40
Rodamientos	625ZZ Ball Bearing	8	1	8
Luces	WS2812b 12 LED Ring (50mm outer diameter)	1	8	8
Pantalla	0.96" OLED I2C Display	2	7	14
Sensores de distancia	HC-SR04 Ultrasonic Sensor	2	3	6
Sensor inercial	GY-521 Gyroscope and Accelerometer	1	4	4
Controladora de motores	PCA9685 16Channel 12Bit PWM Board	1	14	14
Relé	HW-482 / KY-019 5V 10A Relays	1	8	8
Interruptor	19mm Push Button with LED	1	10	10
Conexión externa USB	Micro-USB to DIP Adapter	2	6	12
Transformador principal	XL4016 DC-DC 9A Stepdown Converter	1	14	14
Transformador principal	SZBK07 DC-DC 20A Stepdown Converter	1	16	16
Transformador secundario	LM2596 DC-DC Stepdown Module	1	10	10
Sensor de corriente	ACS712 30A current sensor module & 25V voltage sensor module	1	12	12
Fuente externa	Alimentación externa PC	1	14	14
Batería de litio	6200mAh LiPo 80C 2S	1	35	35
Mando	Logitech Gamepad F310	1	30	30
Conectores Dupont	Pack de conectores Dupont con herramienta crimpadora	1	30	30
Cables 16AWG	Cables 16AWG 20m	1	20	20
Pack de destornillador	Destornillador con cabezales	1	15	15
Pantalla	1,8" TFT with ST7735 w/o SD	1	14	14
Soldador	Soldador de estaño con estaño	1	20	20
Bridas	Bridas de 20cm	1	5	5
Tornillos	M2x8 cylinderhead screws + M2 nuts (100)	1	8	8
Tornillos	M3x8 cylinderhead screws + M3 nuts (100)	2	8	16
Tornillos	M3x20 cylinderhead screws + M3 nuts (100)	1	12	12
Cables, conectores y otros	Dupont bundle	1	20	20
PLA	PLA 1 kg	3	20	60
TPU	TPU 500 g	1	20	20
Amoladora	Amoladora con herramientas para pulido y taladrado	1	30	30

Pelador de cables	Pelador de cables	1	15	15
Impresora 3D	Impresora 3d Ender 3 de creality	1	250	250
Conectores T	Pack de conectores T	1	8	8
Interruptor	Pack de interruptores on/off	1	5	5
Multiplexor I2C	Placa TCA9548A 1-A-8 I2C 8	2	7	14
Conectores de cables	Set de uniones para cables	1	2	2
Terminales para 16AWG	Pack de terminales 16AWG	1	4	4
Controlador principal	Raspberry Pi Zero W	1	30	30
Shield de USB RPi	Conector para USBs de la RPi ZW	1	15	15
Sensores de presión	1kg resistive sensors	4	3	12
Pistola de silicona	Pistola de silicona con silicona	1	20	20
Cargador de batería	Cargador 2s más alargador	1	30	30
Cables USB	Cables USB mini y micro	2	5	10
Cinta para cables	Pack de colores de cinta para cables	1	5	5
Repaciones y total añadido				157
Total devuelto				44
<b>Total proyecto</b>		<b>No incluye el gasto de luz</b>		<b>1285</b>

Tabla 10. Costes finales del desarrollo completo del proyecto (Fuente propia).

---

## Capítulo 7: Trabajos futuros

Dada la naturaleza creativa y de prototipado rápido con gran capacidad de desarrollo, el proyecto cuenta con un gran abanico de trabajos futuros que quedan aquí reflejados:

- Configuración de la RPi ZW con el robot
- Elaboración del código de control a bajo nivel que enlace con el código final para la cinemática inversa con el *IMU-ENCODER*
- Elaboración del código de control de alto nivel
- Elaboración de un código de visión artificial para la cámara
- Control completamente autónomo
- Adaptar el entorno de *PyBullet* al prototipo y componentes
- Partir del modelo para una fase 4 con grandes cambios estructurales
- Teleoperación remota
- Sistema de vigilancia
- ...

---

## Capítulo 8: Bibliografía

- [1] Dave Delisle, “Iron Man Garage Set – Dave’s Geeky Ideas,” 2016. <https://davesgeekyideas.com/2016/04/11/iron-man-garage-set/> (accessed May 29, 2021).
- [2] Boston Dynamics, “Spot,” 2021. [https://www.bostondynamics.com/spot#id\\_first](https://www.bostondynamics.com/spot#id_first) (accessed May 29, 2021).
- [3] BCN3D, “BCN3D MOVEO - Un brazo robótico de código abierto impreso en 3D - BCN3D Technologies.” <https://www.bcn3d.com/es/bcn3d-moveo-un-brazo-robotico-de-codigo-abierto-impreso-en-3d/> (accessed May 29, 2021).
- [4] Niryo, “Niryo One, a 6 axis robot arm designed for education and research.” <https://niryo.com/product/niryo-one/> (accessed May 29, 2021).
- [5] Mark tech, “Hexapod – markwtech.” <https://markwtech.com/robots/hexapod/> (accessed May 29, 2021).
- [6] J. Isabel, “GitHub - JavierIH/miniKame: ESP8266 based quadruped with 8DOF and oscillator based gaits,” *Kame bq labs*, 2016. <https://github.com/JavierIH/miniKame> (accessed May 29, 2021).
- [7] Spot Micro AI, “Home - SpotMicroAI,” 2019. <https://spotmicroai.readthedocs.io/en/latest/> (accessed May 30, 2021).
- [8] M. Kubina, “Github - Spot micro ESP32,” 2019. <https://github.com/michaelkubina/SpotMicroESP32> (accessed Jun. 03, 2021).
- [9] P. Biswal and P. K. Mohanty, “Development of quadruped walking robots: A review,” *Ain Shams Engineering Journal*, vol. 12, no. 2. Ain Shams University, pp. 2017–2031, Jun. 01, 2020, doi: 10.1016/j.asej.2020.11.005.
- [10] F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2. SAGE Publications Inc., Mar. 01, 2019, doi: 10.1177/1729881419839596.
- [11] J. Gonzalez de Santos, Pablo. García, Elena,Estremera, “Quadrupedal Locomotion: An Introduction to the Control of Four-legged Robots - Pablo González de Santos, Elena Garcia, Joaquin Estremera - Google Libros.” [https://books.google.es/books?hl=es&lr=&id=3ZIDAAAQBAJ&oi=fnd&pg=PA3&ots=ZyaGcxcVc7&sig=jDny20frs-PKPIL0DutjFxdJJl&redir\\_esc=y#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=3ZIDAAAQBAJ&oi=fnd&pg=PA3&ots=ZyaGcxcVc7&sig=jDny20frs-PKPIL0DutjFxdJJl&redir_esc=y#v=onepage&q&f=false) (accessed Jun. 06, 2021).
- [12] Honda, “Honda Global I robotics.” <https://global.honda/innovation/robotics.html> (accessed Jun. 06, 2021).
- [13] S. Hirose *et al.*, “Quadruped walking robots at Tokyo Institute of Technology: Design, analysis, and gait control methods,” *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 104–114, 2009, doi: 10.1109/MRA.2009.932524.
- [14] H. Kimura, Y. Fukuoka, and H. Katabuti, “Mechanical Design of a Quadruped ” Tekken 3 & 4 ” and Navigation System Using Laser Range

- Sensor,” 2005.
- [15] Boston Dynamics, “Legacy Robots | Boston Dynamics.” <https://www.bostondynamics.com/legacy> (accessed Jun. 08, 2021).
  - [16] MIT, “‘Blind’ Cheetah 3 robot can climb stairs littered with obstacles | MIT News | Massachusetts Institute of Technology.” <https://news.mit.edu/2018/blind-cheetah-robot-climb-stairs-obstacles-disaster-zones-0705> (accessed Jun. 08, 2021).
  - [17] K. Weinmeister, P. Eckert, H. Witte, and A.-J. Ijspeert, “Cheetah-cub-S: Steering of a quadruped robot using trunk motion,” Mar. 2016, pp. 1–6, doi: 10.1109/ssrr.2015.7443021.
  - [18] National Geographic, “Mini Cheetah, el primer robot acróbata.” [https://www.nationalgeographic.com.es/ciencia/primer-robot-acrobata\\_13973](https://www.nationalgeographic.com.es/ciencia/primer-robot-acrobata_13973) (accessed Jun. 08, 2021).
  - [19] Unitree, “Unitree Robotics - 宇树科技官网.” <https://www.unitree.com/> (accessed Jun. 08, 2021).
  - [20] Ghost robotics, “SOLUTIONS | Ghost Robotics.” <https://www.ghostrobotics.io/partners> (accessed Jun. 08, 2021).
  - [21] “Quadruped Robots Can Climb Ladders Now - IEEE Spectrum.” <https://spectrum.ieee.org/automaton/robotics/robotics-hardware/quadruped-robots-can-climb-ladders-now> (accessed Jun. 08, 2021).
  - [22] “DIY hobby servos quadruped robot | Hackaday.io.” <https://hackaday.io/project/171456-diy-hobby-servos-quadruped-robot> (accessed Jun. 08, 2021).
  - [23] “Exploring the World of Quadruped Robots Post-Acquisition of Boston Dynamics.” <https://singularityhub.com/2014/05/21/now-google-owns-boston-dynamics-youll-have-to-look-elsewhere-for-your-robot-video-fix/> (accessed Jun. 08, 2021).
  - [24] “Italian Quadruped Robot Goes for a Walk - IEEE Spectrum.” <https://spectrum.ieee.org/automaton/robotics/robotics-hardware/italian-quadruped-robot-goes-for-a-walk> (accessed Jun. 08, 2021).
  - [25] Z. G. Zhang and H. Kimura, “Rush: A simple and autonomous quadruped running robot,” *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, vol. 223, no. 3, pp. 323–336, Aug. 2009, doi: 10.1243/09596518JSCE668.
  - [26] “ANYmal - Autonomous Legged Robot | ANYbotics.” <https://www.anybotics.com/anymal-autonomous-legged-robot/> (accessed Jun. 09, 2021).
  - [27] “Quadruped Robot PneuPard Takes Its First Steps - IEEE Spectrum.” <https://spectrum.ieee.org/automaton/robotics/robotics-hardware/japanese-quadruped-robot-pneupard> (accessed Jun. 09, 2021).
  - [28] K. S. Aschenbeck, N. I. Kern, R. J. Bachmann, and R. D. Quinn, “Design of a quadruped robot driven by air muscles,” in *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006, BioRob 2006*, 2006, vol. 2006, pp. 875–880, doi: 10.1109/BIOROB.2006.1639201.
  - [29] N. Tumu, “SpotMicro Balanced - YouTube.”

- <https://www.youtube.com/watch?v=NudqEaxzqT8> (accessed Jun. 12, 2021).
- [30] M. Derese, "SPOTMICRO Walking - YouTube." <https://www.youtube.com/watch?v=4xTkXOf4qvQ> (accessed Jun. 12, 2021).
  - [31] Nao974, "GitHub - Nao974/choreograph-git." <https://github.com/Nao974/choreograph-git> (accessed Jun. 12, 2021).
  - [32] F. Ferri, "Files · master · Custom Robots / SpotMicro - Boston Dynamics Spot inspired robot / Basic - Instructions · GitLab." [https://gitlab.com/custom\\_robots/spotmicroai/basic-instructions/-/tree/master/](https://gitlab.com/custom_robots/spotmicroai/basic-instructions/-/tree/master/) (accessed Jun. 12, 2021).
  - [33] avbotics, "GitHub - avbotics/Spot-Micro-Control-and-Animation: Inverse Kinematics based control and animation of Spotmicro developed in Python for Raspeberry Pi." <https://github.com/avbotics/Spot-Micro-Control-and-Animation> (accessed Jun. 12, 2021).
  - [34] M. Rahme, I. Abraham, M. L. Elwin, and T. D. Murphey, "DR-GMBC." <https://sites.google.com/view/drgmbc> (accessed Jun. 12, 2021).
  - [35] D. J. Hyun, S. Seok, J. Lee, and S. Kim, "High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah," *Int. J. Rob. Res.*, vol. 33, no. 11, pp. 1417–1445, Sep. 2014, doi: 10.1177/0278364914532150.
  - [36] Moribots, "GitHub - OpenQuadruped/spot\_mini\_mini: Dynamics and Domain Randomized Gait Modulation with Bezier Curves for Sim-to-Real Legged Locomotion." [https://github.com/OpenQuadruped/spot\\_mini\\_mini](https://github.com/OpenQuadruped/spot_mini_mini) (accessed Jun. 12, 2021).
  - [37] F. Wilk, "GitHub - FlorianWilk/SpotMicroAI at trottinggait." <https://github.com/FlorianWilk/SpotMicroAI/tree/trottinggait> (accessed Jun. 12, 2021).
  - [38] Mike2192, "GitHub - mike4192/spotMicro: Spot Micro Quadruped Project." <https://github.com/mike4192/spotMicro> (accessed Jun. 12, 2021).
  - [39] J. Tan *et al.*, "Sim-to-Real: Learning Agile Locomotion For Quadruped Robots."
  - [40] X. Meng, S. Wang, Z. Cao, and L. Zhang, "A review of quadruped robots and environment perception," in *Chinese Control Conference, CCC*, Aug. 2016, vol. 2016-August, pp. 6350–6356, doi: 10.1109/ChiCC.2016.7554355.
  - [41] "Gazebo." <http://gazebo-sim.org/> (accessed Jun. 12, 2021).
  - [42] "rviz - ROS Wiki." <http://wiki.ros.org/rviz> (accessed Jun. 12, 2021).
  - [43] "Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics." <https://www.coppeliarobotics.com/> (accessed Jun. 12, 2021).
  - [44] "Autodesk I Software de diseño, ingeniería y construcción en 3D." <https://www.autodesk.es/> (accessed Jun. 12, 2021).
  - [45] "Adams - The Multibody Dynamics Simulation Solution." <https://www.mscsoftware.com/product/adams> (accessed Jun. 12, 2021).
  - [46] "Bullet Real-Time Physics Simulation I Home of Bullet and PyBullet:



- physics simulation for games, visual effects, robotics and reinforcement learning.” <https://pybullet.org/wordpress/> (accessed Jun. 12, 2021).
- [47] “MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink.” [https://es.mathworks.com/?s\\_tid=gn\\_logo](https://es.mathworks.com/?s_tid=gn_logo) (accessed Jun. 12, 2021).
- [48] M. Weyn, “GitHub - maartenweyn/SpotMicro\_ESP32.” [https://github.com/maartenweyn/SpotMicro\\_ESP32](https://github.com/maartenweyn/SpotMicro_ESP32) (accessed Jun. 12, 2021).
- [49] “Best resin 3D printers in 2021 - SLA, DLP, and LCD.” <https://www.aniwaa.com/buyers-guide/3d-printers/the-best-resin-3d-printer-sla-and-dlp/> (accessed Jun. 20, 2021).
- [50] “FDM 3D printing materials compared | Hubs.” <https://www.hubs.com/knowledge-base/fdm-3d-printing-materials-compared/> (accessed Jun. 20, 2021).
- [51] Meperiun3D, “Patrones de relleno: Meperiun3D - Overclock Server.” <https://lordpedal.github.io/3d/patrones-relleno-meperiun3d/> (accessed Jun. 12, 2021).
- [52] Stephan, “TESTING 3D printed INFILL PATTERNS for their STRENGTH - YouTube.” <https://www.youtube.com/watch?v=upELI0HmzHc> (accessed Jun. 12, 2021).
- [53] A. Pandzic, D. Hodzic, and A. Milovanovic, “30TH DAAAM INTERNATIONAL SYMPOSIUM ON INTELLIGENT MANUFACTURING AND AUTOMATION EFFECT OF INFILL TYPE AND DENSITY ON TENSILE PROPERTIES OF PLA MATERIAL FOR FDM PROCESS,” pp. 545–0554, doi: 10.2507/30th.daaam.proceedings.074.
- [54] “What Infill Settings Should I Use for my FDM 3D Print? | SD3D Printing.” <https://www.sd3d.com/infill-settings-fdm-3d-print/> (accessed Jun. 12, 2021).
- [55] “FreeCAD: Your own 3D parametric modeler.” <https://www.freecadweb.org/> (accessed Jun. 13, 2021).
- [56] A. Bouabid, “MG995 Servo Motor | 3D CAD Model Library | GrabCAD.” <https://grabcad.com/library/mg995-servo-motor-2> (accessed Jun. 13, 2021).
- [57] NASA, “Torque (Moment).” <https://www.grc.nasa.gov/www/k-12/airplane/torque.html> (accessed Jun. 13, 2021).
- [58] “Fritzing.” <https://fritzing.org/> (accessed Jun. 14, 2021).
- [59] G. Ismael and P. Carpio, “UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA CARRERA DE INGENIERÍA MECÁNICA.”
- [60] M. M. U. Atique, M. R. I. Sarker, and M. A. R. Ahad, “Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using Denavit-Hartenberg convention,” *Heliyon*, vol. 4, no. 12, p. e01053, Dec. 2018, doi: 10.1016/j.heliyon.2018.e01053.
- [61] D. Ayaklı Robotlar İçin Hidrolik Tahrikli Bir, M. Kalyoncu, M. Arif Sen, and V. Bakircioglu, “Bacak Geliştirilmesi-Development Of a Hydraulically Actuated Leg for Quadruped Robots View project Inverse Kinematic Analysis Of A Quadruped Robot,” *Int. J. Sci. Technol. Res.*, vol. 6, no. 09,

- 2017.
- [62] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Auton. Robots*, vol. 41, no. 1, pp. 259–272, Jan. 2017, doi: 10.1007/s10514-016-9573-1.
  - [63] "01 Umbral.avi - Google Drive." .
  - [64] "Leg Trajectory Planning for Quadruped Robots with ... - MDPIwww.mdpi.com › pdf - Buscar con Google." .
  - [65] H. J. Marcus *et al.*, "Trends in the diffusion of robotic surgery: A retrospective observational study," *Int. J. Med. Robot. + Comput. Assist. Surg.*, vol. 13, no. 4, Dec. 2017, doi: 10.1002/RCS.1870.
  - [66] "Stability analysis and synthesis of statically balanced walking for quadruped robots Freyr Hardarson DAMEK," 2002.
  - [67] S. Tirumala *et al.*, "Learning Stable Manoeuvres in Quadruped Robots from Expert Demonstrations," *29th IEEE Int. Conf. Robot Hum. Interact. Commun. RO-MAN 2020*, pp. 1107–1112, Jul. 2020.
  - [68] Q. Liao, "Review of Quadruped Robots for Dynamic Locomotion," May 2020.
  - [69] C. Liu, Q. Chen, and D. Wang, "CPG-inspired workspace trajectory generation and adaptive locomotion control for Quadruped Robots," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 41, no. 3, pp. 867–880, Jun. 2011, doi: 10.1109/TSMCB.2010.2097589.
  - [70] P. M. Krishna and R. P. Kumar, "Energetics of constant height level bounding in quadruped robots," *Robotica*, vol. 34, no. 2, pp. 403–422, Feb. 2016, doi: 10.1017/S0263574714001532.
  - [71] F. Iida, G. Gómez, and R. Pfeifer, "Exploiting body dynamics for controlling a running quadruped robot," in *2005 International Conference on Advanced Robotics, ICAR '05, Proceedings*, 2005, vol. 2005, pp. 229–235, doi: 10.1109/ICAR.2005.1507417.
  - [72] M. H. Raibert, "Trotting, pacing and bounding by a quadruped robot," *J. Biomech.*, vol. 23, no. SUPPL. 1, pp. 79–98, Jan. 1990, doi: 10.1016/0021-9290(90)90043-3.
  - [73] H. Kimura, Y. Fukuoka, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts," *Int. J. Rob. Res.*, vol. 26, no. 5, pp. 475–490, May 2007, doi: 10.1177/0278364907078089.
  - [74] C. Gehring *et al.*, "Practice Makes Perfect: An Optimization-Based Approach to Controlling Agile Motions for a Quadruped Robot," *IEEE Robot. Autom. Mag.*, vol. 23, no. 1, pp. 34–43, Mar. 2016, doi: 10.1109/MRA.2015.2505910.
  - [75] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, "Optimal Force Distribution for the Legs of a Quadruped Robot\*," 1999.
  - [76] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini, "Passive Whole-Body Control for Quadruped Robots: Experimental Validation over Challenging Terrain," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2553–2560, Jul. 2019, doi: 10.1109/LRA.2019.2908502.

- [77] M. M. Gor, P. M. Pathak, A. K. Samantaray, J. M. Yang, and S. W. Kwak, "Control oriented model-based simulation and experimental studies on a compliant legged quadruped robot," *Rob. Auton. Syst.*, vol. 72, pp. 217–234, Oct. 2015, doi: 10.1016/j.robot.2015.06.001.
- [78] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot," in *IEEE International Conference on Intelligent Robots and Systems*, Dec. 2018, pp. 2245–2252, doi: 10.1109/IROS.2018.8593885.
- [79] J. Li, J. Wang, S. X. Yang, K. Zhou, and H. Tang, "Gait planning and stability control of a quadruped robot," *Comput. Intell. Neurosci.*, vol. 2016, 2016, doi: 10.1155/2016/9853070.
- [80] Z. Li, Q. Ge, W. Ye, and P. Yuan, "Dynamic Balance Optimization and Control of Quadruped Robot Systems with Flexible Joints," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 10, pp. 1338–1351, Oct. 2016, doi: 10.1109/TSMC.2015.2504552.
- [81] H. Chu *et al.*, "Comparative tropism, replication kinetics, and cell damage profiling of SARS-CoV-2 and SARS-CoV with implications for clinical manifestations, transmissibility, and laboratory studies of COVID-19: an observational study," *The Lancet Microbe*, vol. 1, no. 1, pp. e14–e23, May 2020, doi: 10.1016/s2666-5247(20)30004-5.
- [82] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Fast, robust quadruped locomotion over challenging terrain," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 2665–2670, doi: 10.1109/ROBOT.2010.5509805.
- [83] M. Focchi, T. Boaventura, C. Semini, M. Frigerio, J. Buchli, and D. G. Caldwell, "Torque-control based compliant actuation of a quadruped robot," 2012, doi: 10.1109/AMC.2012.6197133.
- [84] Z. Li, S. S. Ge, and S. Liu, "Contact-force distribution optimization and control for quadruped robots using both gradient and adaptive neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 8, pp. 1460–1473, 2014, doi: 10.1109/TNNLS.2013.2293500.
- [85] S. Yi, "Reliable gait planning and control for miniaturized quadruped robot pet," *Mechatronics*, vol. 20, no. 4, pp. 485–495, Jun. 2010, doi: 10.1016/j.mechatronics.2010.04.006.
- [86] Y. Zeng, J. Li, S. X. Yang, and E. Ren, "A bio-inspired control strategy for locomotion of a quadruped robot," *Appl. Sci.*, vol. 8, no. 1, p. 56, Jan. 2018, doi: 10.3390/app8010056.
- [87] H. Kazemi, V. J. Majd, and M. M. Moghaddam, "Modeling and robust backstepping control of an underactuated quadruped robot in bounding motion," *Robotica*, vol. 31, no. 3, pp. 423–439, May 2013, doi: 10.1017/S0263574712000458.
- [88] X. Chen, F. Gao, C. Qi, and X. Tian, "Gait planning for a quadruped robot with one faulty actuator," *Chinese J. Mech. Eng. (English Ed.)*, vol. 28, no. 1, pp. 11–19, Jan. 2015, doi: 10.3901/CJME.2014.1107.167.
- [89] G. Zhang, X. Rong, C. Hui, Y. Li, and B. Li, "Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model

- control,” *Adv. Robot.*, vol. 30, no. 4, pp. 284–297, Feb. 2016, doi: 10.1080/01691864.2015.1113889.
- [90] T. Chen, X. Sun, Z. Xu, Y. Li, X. Rong, and L. Zhou, “A Trot and Flying Trot Control Method for Quadruped Robot Based on Optimal Foot Force Distribution,” *J. Bionic Eng.*, vol. 16, no. 4, pp. 621–632, Jul. 2019, doi: 10.1007/s42235-019-0050-3.
- [91] M. A. ŞEN and M. KALYONCU, “Optimal Tuning of PID Controller Using Grey Wolf Optimizer Algorithm for Quadruped Robot,” *Balk. J. Electr. Comput. Eng.*, vol. 6, no. 1, pp. 29–35, Feb. 2018, doi: 10.17694/bajece.401992.
- [92] M. Arif Sen and M. Kalyoncu, “Grey Wolf Optimizer Based Tuning of a Hybrid LQR-PID Controller for Foot Trajectory Control of a Quadruped Robot,” Jun. 2019.
- [93] P. Sutyasadi and M. Parnichun, “Trotting control of a quadruped robot using PID-ILC,” in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 4400–4405, doi: 10.1109/IECON.2015.7392784.
- [94] K. G. Chae and J. H. Park, “Trajectory optimization with GA and control for quadruped robots,” *J. Mech. Sci. Technol.*, vol. 23, no. 1, pp. 114–123, Jul. 2009, doi: 10.1007/s12206-008-0920-9.
- [95] H. Takemura, M. Deguchi, J. Ueda, Y. Matsumoto, and T. Ogasawara, “Slip-adaptive walk of quadruped robot,” *Rob. Auton. Syst.*, vol. 53, no. 2, pp. 124–141, Nov. 2005, doi: 10.1016/j.robot.2005.07.002.
- [96] T. Kano, K. Nagasawa, D. Owaki, A. Tero, and A. Ishiguro, “A CPG-based decentralized control of a quadruped robot inspired by true slime mold,” in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, pp. 4928–4933, doi: 10.1109/IROS.2010.5650318.
- [97] S. Suzuki, T. Kano, A. J. Ijspeert, and A. Ishiguro, “Sprawling Quadruped Robot Driven by Decentralized Control With Cross-Coupled Sensory Feedback Between Legs and Trunk,” *Front. Neurobot.*, vol. 14, p. 607455, Jan. 2021, doi: 10.3389/fnbot.2020.607455.
- [98] “Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds I Papers With Code.” .
- [99] “Stanford Doggo: An Open-Source, Quasi-Direct-Drive Quadruped I Papers With Code.” .
- [100] “Evolved embodied phase coordination enables robust quadruped robot locomotion I Papers With Code.” .
- [101] T. F. Nygaard, C. P. Martin, J. Torresen, K. Glette, and D. Howard, “Real-world embodied AI through a morphologically adaptive quadruped robot,” *Nat. Mach. Intell.*, vol. 3, no. 5, pp. 410–419, May 2021, doi: 10.1038/s42256-021-00320-3.
- [102] C. Lee and D. An, “Reinforcement learning and neural network-based artificial intelligence control algorithm for self-balancing quadruped robot,” *J. Mech. Sci. Technol.*, vol. 35, no. 1, pp. 307–322, Jan. 2021, doi: 10.1007/s12206-020-1230-0.

- [103] M. Focchi *et al.*, “An Optimization-Based Locomotion Controller for Quadruped Robots Leveraging Cartesian Impedance Control,” *Front. Robot. AI*, vol. 7, p. 48, 2020, doi: 10.3389/frobt.2020.00048.
- [104] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, “Path tracking based on closed-loop control for a quadruped robot in a cluttered environment,” *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 124, no. 2, pp. 272–280, Jun. 2002, doi: 10.1115/1.1472460.
- [105] K. Otsu, A. Ishiguro, A. Fujii, T. Aoki, and P. Eggenberger, “Evolving an adaptive controller for a quadruped-robot with dynamically-rearranging neural networks,” in *IEEE International Conference on Intelligent Robots and Systems*, 2001, vol. 4, pp. 2036–2044, doi: 10.1109/iros.2001.976372.
- [106] “Emergent Real-World Robotic Skills via Unsupervised Off-Policy Reinforcement Learning I Papers With Code.” .
- [107] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, “Learning to Walk via Deep Reinforcement Learning.”
- [108] B. Na, H. Choi, and K. Kong, “Design of a direct-driven linear actuator for a high-speed quadruped robot, cheetaroid-I,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 924–933, Apr. 2015, doi: 10.1109/TMECH.2014.2326696.
- [109] T. Sun, X. Xiong, Z. Dai, and P. Manoonpong, “Small-Sized Reconfigurable Quadruped Robot With Multiple Sensory Feedback for Studying Adaptive and Versatile Behaviors,” *Front. Neurorobot.*, vol. 14, p. 14, Feb. 2020, doi: 10.3389/fnbot.2020.00014.
- [110] G. Chen, S. Guo, B. Hou, and J. Wang, “Virtual Model Control for Quadruped Robots,” *IEEE Access*, vol. 8, pp. 140736–140751, 2020, doi: 10.1109/ACCESS.2020.3013434.
- [111] “NeuralSim: Augmenting Differentiable Simulators with Neural Networks I Papers With Code.” .

---

## Anexo 1: Enlaces a los códigos, programas, multimedia y explicaciones

En este anexo se encuentran los enlaces a todos los materiales referenciados a lo largo del documento de forma directa a sus carpetas contenedoras para mayor conveniencia:

- [Carpeta general:](#) contiene todo el trabajo final de grado.
- [Carpeta de los modelos 3D:](#) contiene las subcarpetas para los ensamblajes y modelos 3D para todas y cada una de las fases.
- [Carpeta de multimedia general:](#) contiene las subcarpetas para los ensamblajes, modelos 3D para todas y cada una de las fases, estudios en inventor, limitación de grados de libertad y vídeos de pruebas e imágenes de los distintos procesos y casos.
- [Carpeta de códigos de prueba:](#) contiene todos los códigos de Arduino empleados para verificar y calibrar los componentes, así como las listas y librerías empleadas para todos los entornos y pasos.
- [Carpeta de códigos finales:](#) contiene los códigos de la versión final, así como los agentes entrenados.
- [Carpeta de explicaciones:](#) contiene los vídeos explicativos mencionados a lo largo de la memoria y que sustituyen al uso de centenares de imágenes y texto.
- [Carpeta de documentación:](#) contiene el informe de costes, los estudios en Autodesk inventor, el resumen y esta memoria.
- [Carpeta de circuitos:](#) contiene todas las versiones de los circuitos diseñados e implementados.
- [Carpeta de los perfiles de impresión:](#) contiene las configuraciones usadas para Ultimaker Cura.

---

## Anexo 2: Ampliación del estado del arte

En este anexo se encuentran los enlaces y títulos de todos aquellos artículos que, aún no habiendo sido citados o mencionados, considero pueden ser importantes para el desarrollo de plataformas similares y que fueron consultados y leídos antes de centrar el capítulo 3 exclusivamente en las plataformas de spot micro y no en las generales.

### Sistemas de control de robots tradicionales

1. Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using Denavit-Hartenberg convention [60]
2. Inverse Kinematic Analysis Of A Quadruped Robot [61]
3. High-slope terrain locomotion for torque-controlled quadruped robots (AURO'17) [62]
4. Control inicial del chetah [63]
5. Leg Trajectory Planning for Quadruped Robots with High-Speed Trot Gait [64]
6. Analysis and research of quadruped robot's legs: A comprehensive review [65]
7. Stability analysis and synthesis of statically balanced walking for quadruped robots [66]
8. Learning Stable Manoeuvres in Quadruped Robots from Expert Demonstrations [67]
9. Review of Quadruped Robots for Dynamic Locomotion [68]
10. CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots [69]
11. Energetics of constant height level bounding in quadruped robots [70]
12. Exploiting body dynamics for controlling a running quadruped robot [71]
13. Trotting, pacing and bounding by a quadruped robot [72]
14. Adaptive Dynamic Walking of a Quadruped Robot on Natural Ground Based on Biological Concepts [73]
15. Practice Makes Perfect: An Optimization-Based Approach to Controlling Agile Motions for a Quadruped Robot [74]
16. Optimal Force Distribution for the Legs of a Quadruped Robot [75]
17. Passive Whole-Body Control for Quadruped Robots: Experimental Validation Over Challenging Terrain [76]
18. Control oriented model-based simulation and experimental studies on a compliant legged quadruped robot [77]

19. MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot [78]
20. Gait Planning and Stability Control of a Quadruped Robot [79]
21. Dynamic Balance Optimization and Control of Quadruped Robot Systems With Flexible Joints [80]
22. Gait Adaptation in a Quadruped Robot [81]
23. Fast, robust quadruped locomotion over challenging terrain [82]
24. Torque-control based compliant actuation of a quadruped robot [83]
25. Contact-Force Distribution Optimization and Control for Quadruped Robots Using Both Gradient and Adaptive Neural Networks [84]
26. Reliable gait planning and control for miniaturized quadruped robot pet [85]
27. A Bio-Inspired Control Strategy for Locomotion of a Quadruped Robot [86]
28. Modeling and robust backstepping control of an underactuated quadruped robot in bounding motion [87]
29. Gait planning for a quadruped robot with one faulty actuator [88]
30. Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control [89]
31. A trot and flying trot control method for quadruped robot based on optimal foot force distribution [90]
32. Optimal Tuning of PID Controller Using Grey Wolf Optimizer Algorithm for Quadruped Robot [91]
33. Grey Wolf Optimizer Based Tuning of a Hybrid LQR-PID Controller for Foot Trajectory Control of a Quadruped Robot [92]
34. Trotting control of a quadruped robot using PID-ILC [93]
35. Trajectory optimization with GA and control for quadruped robots [94]
36. Slip-adaptive walk of quadruped robot [95]
37. A CPG-based decentralized control of a quadruped robot inspired by true slime mold [96]
38. Sprawling Quadruped Robot Driven by Decentralized Control With Cross-Coupled Sensory Feedback Between Legs and Trunk [97]
39. Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds [98]
40. Stanford Doggo: An Open-Source, Quasi-Direct-Drive Quadruped [99]
41. Evolved embodied phase coordination enables robust quadruped robot locomotion [100]

Sistemas de control con algoritmos y redes neuronales aplicadas



42. Real-world embodied AI through a morphologically adaptive quadruped robot [101]
43. Reinforcement learning and neural network-based artificial intelligence control algorithm for self-balancing quadruped robot [102]
44. An Optimization-Based Locomotion Controller for Quadruped Robots Leveraging Cartesian Impedance Control [103]
45. Path Tracking Based on Closed-Loop Control for a Quadruped Robot in a Cluttered Environment [104]
46. Evolving an adaptive controller for a quadruped-robot with dynamically-rearranging neural networks [105]
47. Emergent Real-World Robotic Skills via Unsupervised Off-Policy Reinforcement Learning [106]
48. Learning to Walk via Deep Reinforcement Learning [107]
49. Linear Policies are Sufficient to Enable Low-cost Quadrupedal Robots to Traverse Rough Terrain [34]

## Sensores y actuadores en robots cuadrúpedos

50. Design of a Direct-Driven Linear Actuator for a High-Speed Quadruped Robot, Cheetaroid-I [108]
51. A review of quadruped robots and environment perception [40]
52. Small-Sized Reconfigurable Quadruped Robot With Multiple Sensory Feedback for Studying Adaptive and Versatile Behaviors [109]

## Sistemas de simulación y tele operación

53. Virtual Model Control for Quadruped Robots [110]
54. NeuralSim: Augmenting Differentiable Simulators with Neural Networks [111]

## Bibliografía de los artículos de investigación arriba mencionados:

- [1] Dave Delisle, "Iron Man Garage Set – Dave's Geeky Ideas," 2016. <https://davesgeekyideas.com/2016/04/11/iron-man-garage-set/> (accessed May 29, 2021).
- [2] Boston Dynamics, "Spot," 2021. [https://www.bostondynamics.com/spot#id\\_first](https://www.bostondynamics.com/spot#id_first) (accessed May 29, 2021).
- [3] BCN3D, "BCN3D MOVEO - Un brazo robótico de código abierto impreso en 3D - BCN3D Technologies." <https://www.bcn3d.com/es/bcn3d-moveo-203>

- un-brazo-robotico-de-codigo-abierto-impreso-en-3d/ (accessed May 29, 2021).
- [4] Niryo, "Niryo One, a 6 axis robot arm designed for education and research." <https://niryo.com/product/niryo-one/> (accessed May 29, 2021).
  - [5] Mark tech, "Hexapod – markwtech." <https://markwtech.com/robots/hexapod/> (accessed May 29, 2021).
  - [6] J. Isabel, "GitHub - JavierIH/miniKame: ESP8266 based quadruped with 8DOF and oscillator based gaits.," *Kame bq labs*, 2016. <https://github.com/JavierIH/miniKame> (accessed May 29, 2021).
  - [7] Spot Micro AI, "Home - SpotMicroAI," 2019. <https://spotmicroai.readthedocs.io/en/latest/> (accessed May 30, 2021).
  - [8] M. Kubina, "Github - Spot micro ESP32," 2019. <https://github.com/michaelkubina/SpotMicroESP32> (accessed Jun. 03, 2021).
  - [9] P. Biswal and P. K. Mohanty, "Development of quadruped walking robots: A review," *Ain Shams Engineering Journal*, vol. 12, no. 2. Ain Shams University, pp. 2017–2031, Jun. 01, 2020, doi: 10.1016/j.asej.2020.11.005.
  - [10] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2. SAGE Publications Inc., Mar. 01, 2019, doi: 10.1177/1729881419839596.
  - [11] J. Gonzalez de Santos, Pablo. García, Elena,Estremera, "Quadrupedal Locomotion: An Introduction to the Control of Four-legged Robots - Pablo González de Santos, Elena Garcia, Joaquin Estremera - Google Libros." [https://books.google.es/books?hl=es&lr=&id=3ZIDAAAQBAJ&oi=fnd&pg=PA3&ots=ZyaGcxcVc7&sig=jDny20frs-PKPIL0DutjFxdJJII&redir\\_esc=y#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=3ZIDAAAQBAJ&oi=fnd&pg=PA3&ots=ZyaGcxcVc7&sig=jDny20frs-PKPIL0DutjFxdJJII&redir_esc=y#v=onepage&q&f=false) (accessed Jun. 06, 2021).
  - [12] Honda, "Honda Global I robotics." <https://global.honda/innovation/robotics.html> (accessed Jun. 06, 2021).
  - [13] S. Hirose *et al.*, "Quadruped walking robots at Tokyo Institute of Technology: Design, analysis, and gait control methods," *IEEE Robot. Autom. Mag.*, vol. 16, no. 2, pp. 104–114, 2009, doi: 10.1109/MRA.2009.932524.
  - [14] H. Kimura, Y. Fukuoka, and H. Katabuti, "Mechanical Design of a Quadruped " Tekken 3 & 4 " and Navigation System Using Laser Range Sensor," 2005.
  - [15] Boston Dynamics, "Legacy Robots I Boston Dynamics." <https://www.bostondynamics.com/legacy> (accessed Jun. 08, 2021).
  - [16] MIT, "'Blind' Cheetah 3 robot can climb stairs littered with obstacles I MIT News I Massachusetts Institute of Technology." <https://news.mit.edu/2018/blind-cheetah-robot-climb-stairs-obstacles-disaster-zones-0705> (accessed Jun. 08, 2021).
  - [17] K. Weinmeister, P. Eckert, H. Witte, and A.-J. Ijspeert, "Cheetah-cub-S: Steering of a quadruped robot using trunk motion," Mar. 2016, pp. 1–6, doi: 10.1109/ssrr.2015.7443021.

- [18] National Geographic, “Mini Cheetah, el primer robot acróbata.” [https://www.nationalgeographic.com.es/ciencia/primer-robot-acrobata\\_13973](https://www.nationalgeographic.com.es/ciencia/primer-robot-acrobata_13973) (accessed Jun. 08, 2021).
- [19] Unitree, “Unitree Robotics - 宇树科技官网.” <https://www.unitree.com/> (accessed Jun. 08, 2021).
- [20] Ghost robotics, “SOLUTIONS | Ghost Robotics.” <https://www.ghostrobotics.io/partners> (accessed Jun. 08, 2021).
- [21] “Quadruped Robots Can Climb Ladders Now - IEEE Spectrum.” <https://spectrum.ieee.org/automaton/robotics/robotics-hardware/quadruped-robots-can-climb-ladders-now> (accessed Jun. 08, 2021).
- [22] “DIY hobby servos quadruped robot | Hackaday.io.” <https://hackaday.io/project/171456-diy-hobby-servos-quadruped-robot> (accessed Jun. 08, 2021).
- [23] “Exploring the World of Quadruped Robots Post-Acquisition of Boston Dynamics.” <https://singularityhub.com/2014/05/21/now-google-owns-boston-dynamics-youll-have-to-look-elsewhere-for-your-robot-video-fix/> (accessed Jun. 08, 2021).
- [24] “Italian Quadruped Robot Goes for a Walk - IEEE Spectrum.” <https://spectrum.ieee.org/automaton/robotics/robotics-hardware/italian-quadruped-robot-goes-for-a-walk> (accessed Jun. 08, 2021).
- [25] Z. G. Zhang and H. Kimura, “Rush: A simple and autonomous quadruped running robot,” *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, vol. 223, no. 3, pp. 323–336, Aug. 2009, doi: 10.1243/09596518JSCE668.
- [26] “ANYmal - Autonomous Legged Robot | ANYbotics.” <https://www.anybotics.com/anymal-autonomous-legged-robot/> (accessed Jun. 09, 2021).
- [27] “Quadruped Robot PneuPard Takes Its First Steps - IEEE Spectrum.” <https://spectrum.ieee.org/automaton/robotics/robotics-hardware/japanese-quadruped-robot-pneupard> (accessed Jun. 09, 2021).
- [28] K. S. Aschenbeck, N. I. Kern, R. J. Bachmann, and R. D. Quinn, “Design of a quadruped robot driven by air muscles,” in *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006, BioRob 2006*, 2006, vol. 2006, pp. 875–880, doi: 10.1109/BIOROB.2006.1639201.
- [29] N. Tumu, “SpotMicro Balanced - YouTube.” <https://www.youtube.com/watch?v=NudqEaxzqT8> (accessed Jun. 12, 2021).
- [30] M. Derese, “SPOTMICRO Walking - YouTube.” <https://www.youtube.com/watch?v=4xTkXOf4qvQ> (accessed Jun. 12, 2021).
- [31] Nao974, “GitHub - Nao974/choreograph-git.” <https://github.com/Nao974/choreograph-git> (accessed Jun. 12, 2021).
- [32] F. Ferri, “Files · master · Custom Robots / SpotMicro - Boston Dynamics Spot inspired robot / Basic - Instructions · GitLab.” [https://gitlab.com/custom\\_robots/spotmicroai/basic-instructions/](https://gitlab.com/custom_robots/spotmicroai/basic-instructions/)

- /tree/master/ (accessed Jun. 12, 2021).
- [33] avbotics, "GitHub - avbotics/Spot-Micro-Control-and-Animation: Inverse Kinematics based control and animation of Spotmicro developed in Python for Raspeberry Pi." <https://github.com/avbotics/Spot-Micro-Control-and-Animation> (accessed Jun. 12, 2021).
  - [34] M. Rahme, I. Abraham, M. L. Elwin, and T. D. Murphey, "DR-GMBC." <https://sites.google.com/view/drgmbc> (accessed Jun. 12, 2021).
  - [35] D. J. Hyun, S. Seok, J. Lee, and S. Kim, "High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah," *Int. J. Rob. Res.*, vol. 33, no. 11, pp. 1417–1445, Sep. 2014, doi: 10.1177/0278364914532150.
  - [36] Moribots, "GitHub - OpenQuadruped/spot\_mini\_mini: Dynamics and Domain Randomized Gait Modulation with Bezier Curves for Sim-to-Real Legged Locomotion." [https://github.com/OpenQuadruped/spot\\_mini\\_mini](https://github.com/OpenQuadruped/spot_mini_mini) (accessed Jun. 12, 2021).
  - [37] F. Wilk, "GitHub - FlorianWilk/SpotMicroAI at trottinggait." <https://github.com/FlorianWilk/SpotMicroAI/tree/trottinggait> (accessed Jun. 12, 2021).
  - [38] Mike2192, "GitHub - mike4192/spotMicro: Spot Micro Quadruped Project." <https://github.com/mike4192/spotMicro> (accessed Jun. 12, 2021).
  - [39] J. Tan *et al.*, "Sim-to-Real: Learning Agile Locomotion For Quadruped Robots."
  - [40] X. Meng, S. Wang, Z. Cao, and L. Zhang, "A review of quadruped robots and environment perception," in *Chinese Control Conference, CCC*, Aug. 2016, vol. 2016-August, pp. 6350–6356, doi: 10.1109/ChiCC.2016.7554355.
  - [41] "Gazebo." <http://gazebo.org/> (accessed Jun. 12, 2021).
  - [42] "rviz - ROS Wiki." <http://wiki.ros.org/rviz> (accessed Jun. 12, 2021).
  - [43] "Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics." <https://www.coppeliarobotics.com/> (accessed Jun. 12, 2021).
  - [44] "Autodesk I Software de diseño, ingeniería y construcción en 3D." <https://www.autodesk.es/> (accessed Jun. 12, 2021).
  - [45] "Adams - The Multibody Dynamics Simulation Solution." <https://www.mscsoftware.com/product/adams> (accessed Jun. 12, 2021).
  - [46] "Bullet Real-Time Physics Simulation I Home of Bullet and PyBullet: physics simulation for games, visual effects, robotics and reinforcement learning." <https://pybullet.org/wordpress/> (accessed Jun. 12, 2021).
  - [47] "MathWorks - Creadores de MATLAB y Simulink - MATLAB y Simulink - MATLAB & Simulink." [https://es.mathworks.com/?s\\_tid=gn\\_logo](https://es.mathworks.com/?s_tid=gn_logo) (accessed Jun. 12, 2021).
  - [48] M. Weyn, "GitHub - maartenweyn/SpotMicro\_ESP32." [https://github.com/maartenweyn/SpotMicro\\_ESP32](https://github.com/maartenweyn/SpotMicro_ESP32) (accessed Jun. 12, 2021).
  - [49] "Best resin 3D printers in 2021 - SLA, DLP, and LCD." <https://www.aniwaa.com/buyers-guide/3d-printers/the-best-resin-3d->

- printer-sla-and-dlp/ (accessed Jun. 20, 2021).
- [50] “FDM 3D printing materials compared | Hubs.” <https://www.hubs.com/knowledge-base/fdm-3d-printing-materials-compared/> (accessed Jun. 20, 2021).
  - [51] Meperiun3D, “Patrones de relleno: Meperiun3D - Overclock Server.” <https://lordpedal.github.io/3d/patrones-relleno-meperiun3d/> (accessed Jun. 12, 2021).
  - [52] Stephan, “TESTING 3D printed INFILL PATTERNS for their STRENGTH - YouTube.” <https://www.youtube.com/watch?v=upELI0HmzHc> (accessed Jun. 12, 2021).
  - [53] A. Pandzic, D. Hodzic, and A. Milovanovic, “30TH DAAAM INTERNATIONAL SYMPOSIUM ON INTELLIGENT MANUFACTURING AND AUTOMATION EFFECT OF INFILL TYPE AND DENSITY ON TENSILE PROPERTIES OF PLA MATERIAL FOR FDM PROCESS,” pp. 545–0554, doi: 10.2507/30th.daaam.proceedings.074.
  - [54] “What Infill Settings Should I Use for my FDM 3D Print? | SD3D Printing.” <https://www.sd3d.com/infill-settings-fdm-3d-print/> (accessed Jun. 12, 2021).
  - [55] “FreeCAD: Your own 3D parametric modeler.” <https://www.freecadweb.org/> (accessed Jun. 13, 2021).
  - [56] A. Bouabid, “MG995 Servo Motor | 3D CAD Model Library | GrabCAD.” <https://grabcad.com/library/mg995-servo-motor-2> (accessed Jun. 13, 2021).
  - [57] NASA, “Torque (Moment).” <https://www.grc.nasa.gov/www/k-12/airplane/torque.html> (accessed Jun. 13, 2021).
  - [58] “Fritzing.” <https://fritzing.org/> (accessed Jun. 14, 2021).
  - [59] G. Ismael and P. Carpio, “UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA CARRERA DE INGENIERÍA MECÁNICA.”
  - [60] M. M. U. Atique, M. R. I. Sarker, and M. A. R. Ahad, “Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using Denavit-Hartenberg convention,” *Heliyon*, vol. 4, no. 12, p. e01053, Dec. 2018, doi: 10.1016/j.heliyon.2018.e01053.
  - [61] D. Ayaklı Robotlar İçin Hidrolik Tahrikli Bir, M. Kalyoncu, M. Arif Sen, and V. Bakircioglu, “Bacak Geliştirilmesi-Development Of a Hydraulically Actuated Leg for Quadruped Robots View project Inverse Kinematic Analysis Of A Quadruped Robot,” *Int. J. Sci. Technol. Res.*, vol. 6, no. 09, 2017.
  - [62] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, “High-slope terrain locomotion for torque-controlled quadruped robots,” *Auton. Robots*, vol. 41, no. 1, pp. 259–272, Jan. 2017, doi: 10.1007/s10514-016-9573-1.
  - [63] “01 Umbral.avi - Google Drive.”
  - [64] “Leg Trajectory Planning for Quadruped Robots with ... - MDPI [www.mdpi.com](http://www.mdpi.com) › pdf - Buscar con Google.”
  - [65] H. J. Marcus *et al.*, “Trends in the diffusion of robotic surgery: A retrospective observational study,” *Int. J. Med. Robot. + Comput. Assist.*

- Surg.*, vol. 13, no. 4, Dec. 2017, doi: 10.1002/RCS.1870.
- [66] “Stability analysis and synthesis of statically balanced walking for quadruped robots Freyr Hardarson DAMEK,” 2002.
  - [67] S. Tirumala *et al.*, “Learning Stable Manoeuvres in Quadruped Robots from Expert Demonstrations,” *29th IEEE Int. Conf. Robot Hum. Interact. Commun. RO-MAN 2020*, pp. 1107–1112, Jul. 2020.
  - [68] Q. Liao, “Review of Quadruped Robots for Dynamic Locomotion,” May 2020.
  - [69] C. Liu, Q. Chen, and D. Wang, “CPG-inspired workspace trajectory generation and adaptive locomotion control for Quadruped Robots,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 41, no. 3, pp. 867–880, Jun. 2011, doi: 10.1109/TSMCB.2010.2097589.
  - [70] P. M. Krishna and R. P. Kumar, “Energetics of constant height level bounding in quadruped robots,” *Robotica*, vol. 34, no. 2, pp. 403–422, Feb. 2016, doi: 10.1017/S0263574714001532.
  - [71] F. Iida, G. Gómez, and R. Pfeifer, “Exploiting body dynamics for controlling a running quadruped robot,” in *2005 International Conference on Advanced Robotics, ICAR '05, Proceedings*, 2005, vol. 2005, pp. 229–235, doi: 10.1109/ICAR.2005.1507417.
  - [72] M. H. Raibert, “Trotting, pacing and bounding by a quadruped robot,” *J. Biomech.*, vol. 23, no. SUPPL. 1, pp. 79–98, Jan. 1990, doi: 10.1016/0021-9290(90)90043-3.
  - [73] H. Kimura, Y. Fukuoka, and A. H. Cohen, “Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts,” *Int. J. Rob. Res.*, vol. 26, no. 5, pp. 475–490, May 2007, doi: 10.1177/0278364907078089.
  - [74] C. Gehring *et al.*, “Practice Makes Perfect: An Optimization-Based Approach to Controlling Agile Motions for a Quadruped Robot,” *IEEE Robot. Autom. Mag.*, vol. 23, no. 1, pp. 34–43, Mar. 2016, doi: 10.1109/MRA.2015.2505910.
  - [75] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, “Optimal Force Distribution for the Legs of a Quadruped Robot\*,” 1999.
  - [76] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini, “Passive Whole-Body Control for Quadruped Robots: Experimental Validation over Challenging Terrain,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2553–2560, Jul. 2019, doi: 10.1109/LRA.2019.2908502.
  - [77] M. M. Gor, P. M. Pathak, A. K. Samantaray, J. M. Yang, and S. W. Kwak, “Control oriented model-based simulation and experimental studies on a compliant legged quadruped robot,” *Rob. Auton. Syst.*, vol. 72, pp. 217–234, Oct. 2015, doi: 10.1016/j.robot.2015.06.001.
  - [78] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot,” in *IEEE International Conference on Intelligent Robots and Systems*, Dec. 2018, pp. 2245–2252, doi: 10.1109/IROS.2018.8593885.
  - [79] J. Li, J. Wang, S. X. Yang, K. Zhou, and H. Tang, “Gait planning and stability control of a quadruped robot,” *Comput. Intell. Neurosci.*, vol. 2016,

- 2016, doi: 10.1155/2016/9853070.
- [80] Z. Li, Q. Ge, W. Ye, and P. Yuan, "Dynamic Balance Optimization and Control of Quadraped Robot Systems with Flexible Joints," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 10, pp. 1338–1351, Oct. 2016, doi: 10.1109/TSMC.2015.2504552.
  - [81] H. Chu *et al.*, "Comparative tropism, replication kinetics, and cell damage profiling of SARS-CoV-2 and SARS-CoV with implications for clinical manifestations, transmissibility, and laboratory studies of COVID-19: an observational study," *The Lancet Microbe*, vol. 1, no. 1, pp. e14–e23, May 2020, doi: 10.1016/s2666-5247(20)30004-5.
  - [82] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Fast, robust quadraped locomotion over challenging terrain," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 2665–2670, doi: 10.1109/ROBOT.2010.5509805.
  - [83] M. Focchi, T. Boaventura, C. Semini, M. Frigerio, J. Buchli, and D. G. Caldwell, "Torque-control based compliant actuation of a quadraped robot," 2012, doi: 10.1109/AMC.2012.6197133.
  - [84] Z. Li, S. S. Ge, and S. Liu, "Contact-force distribution optimization and control for quadraped robots using both gradient and adaptive neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 8, pp. 1460–1473, 2014, doi: 10.1109/TNNLS.2013.2293500.
  - [85] S. Yi, "Reliable gait planning and control for miniaturized quadraped robot pet," *Mechatronics*, vol. 20, no. 4, pp. 485–495, Jun. 2010, doi: 10.1016/j.mechatronics.2010.04.006.
  - [86] Y. Zeng, J. Li, S. X. Yang, and E. Ren, "A bio-inspired control strategy for locomotion of a quadraped robot," *Appl. Sci.*, vol. 8, no. 1, p. 56, Jan. 2018, doi: 10.3390/app8010056.
  - [87] H. Kazemi, V. J. Majd, and M. M. Moghaddam, "Modeling and robust backstepping control of an underactuated quadraped robot in bounding motion," *Robotica*, vol. 31, no. 3, pp. 423–439, May 2013, doi: 10.1017/S0263574712000458.
  - [88] X. Chen, F. Gao, C. Qi, and X. Tian, "Gait planning for a quadraped robot with one faulty actuator," *Chinese J. Mech. Eng. (English Ed.)*, vol. 28, no. 1, pp. 11–19, Jan. 2015, doi: 10.3901/CJME.2014.1107.167.
  - [89] G. Zhang, X. Rong, C. Hui, Y. Li, and B. Li, "Torso motion control and toe trajectory generation of a trotting quadraped robot based on virtual model control," *Adv. Robot.*, vol. 30, no. 4, pp. 284–297, Feb. 2016, doi: 10.1080/01691864.2015.1113889.
  - [90] T. Chen, X. Sun, Z. Xu, Y. Li, X. Rong, and L. Zhou, "A Trot and Flying Trot Control Method for Quadraped Robot Based on Optimal Foot Force Distribution," *J. Bionic Eng.*, vol. 16, no. 4, pp. 621–632, Jul. 2019, doi: 10.1007/s42235-019-0050-3.
  - [91] M. A. ŞEN and M. KALYONCU, "Optimal Tuning of PID Controller Using Grey Wolf Optimizer Algorithm for Quadraped Robot," *Balk. J. Electr. Comput. Eng.*, vol. 6, no. 1, pp. 29–35, Feb. 2018, doi: 10.17694/bajece.401992.

- [92] M. Arif Sen and M. Kalyoncu, "Grey Wolf Optimizer Based Tuning of a Hybrid LQR-PID Controller for Foot Trajectory Control of a Quadruped Robot," Jun. 2019.
- [93] P. Sutyasadi and M. Parnichun, "Trotting control of a quadruped robot using PID-ILC," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 4400–4405, doi: 10.1109/IECON.2015.7392784.
- [94] K. G. Chae and J. H. Park, "Trajectory optimization with GA and control for quadruped robots," *J. Mech. Sci. Technol.*, vol. 23, no. 1, pp. 114–123, Jul. 2009, doi: 10.1007/s12206-008-0920-9.
- [95] H. Takemura, M. Deguchi, J. Ueda, Y. Matsumoto, and T. Ogasawara, "Slip-adaptive walk of quadruped robot," *Rob. Auton. Syst.*, vol. 53, no. 2, pp. 124–141, Nov. 2005, doi: 10.1016/j.robot.2005.07.002.
- [96] T. Kano, K. Nagasawa, D. Owaki, A. Tero, and A. Ishiguro, "A CPG-based decentralized control of a quadruped robot inspired by true slime mold," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, pp. 4928–4933, doi: 10.1109/IROS.2010.5650318.
- [97] S. Suzuki, T. Kano, A. J. Ijspeert, and A. Ishiguro, "Sprawling Quadruped Robot Driven by Decentralized Control With Cross-Coupled Sensory Feedback Between Legs and Trunk," *Front. Neurobot.*, vol. 14, p. 607455, Jan. 2021, doi: 10.3389/fnbot.2020.607455.
- [98] "Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds I Papers With Code." .
- [99] "Stanford Doggo: An Open-Source, Quasi-Direct-Drive Quadruped I Papers With Code." .
- [100] "Evolved embodied phase coordination enables robust quadruped robot locomotion I Papers With Code." .
- [101] T. F. Nygaard, C. P. Martin, J. Torresen, K. Glette, and D. Howard, "Real-world embodied AI through a morphologically adaptive quadruped robot," *Nat. Mach. Intell.*, vol. 3, no. 5, pp. 410–419, May 2021, doi: 10.1038/s42256-021-00320-3.
- [102] C. Lee and D. An, "Reinforcement learning and neural network-based artificial intelligence control algorithm for self-balancing quadruped robot," *J. Mech. Sci. Technol.*, vol. 35, no. 1, pp. 307–322, Jan. 2021, doi: 10.1007/s12206-020-1230-0.
- [103] M. Focchi *et al.*, "An Optimization-Based Locomotion Controller for Quadruped Robots Leveraging Cartesian Impedance Control," *Front. Robot. AI*, vol. 7, p. 48, 2020, doi: 10.3389/frobt.2020.00048.
- [104] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, "Path tracking based on closed-loop control for a quadruped robot in a cluttered environment," *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 124, no. 2, pp. 272–280, Jun. 2002, doi: 10.1115/1.1472460.
- [105] K. Otsu, A. Ishiguro, A. Fujii, T. Aoki, and P. Eggenberger, "Evolving an adaptive controller for a quadruped-robot with dynamically-rearranging neural networks," in *IEEE International Conference on Intelligent Robots*



- and Systems*, 2001, vol. 4, pp. 2036–2044, doi: 10.1109/iros.2001.976372.
- [106] “Emergent Real-World Robotic Skills via Unsupervised Off-Policy Reinforcement Learning I Papers With Code.” .
  - [107] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, “Learning to Walk via Deep Reinforcement Learning.”
  - [108] B. Na, H. Choi, and K. Kong, “Design of a direct-driven linear actuator for a high-speed quadruped robot, cheetaroid-I,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 924–933, Apr. 2015, doi: 10.1109/TMECH.2014.2326696.
  - [109] T. Sun, X. Xiong, Z. Dai, and P. Manoonpong, “Small-Sized Reconfigurable Quadruped Robot With Multiple Sensory Feedback for Studying Adaptive and Versatile Behaviors,” *Front. Neurorobot.*, vol. 14, p. 14, Feb. 2020, doi: 10.3389/fnbot.2020.00014.
  - [110] G. Chen, S. Guo, B. Hou, and J. Wang, “Virtual Model Control for Quadruped Robots,” *IEEE Access*, vol. 8, pp. 140736–140751, 2020, doi: 10.1109/ACCESS.2020.3013434.
  - [111] “NeuralSim: Augmenting Differentiable Simulators with Neural Networks I Papers With Code.” .

---

## Anexo 3: Impresión 3D

Una parte fundamental para el desarrollo de este trabajo ha sido la impresión 3D. Ha habido incidentes y aprendizajes diversos, así como una curva de aprendizaje para el proceso. Para poder visualizar el vídeo en el que se explica todo lo relacionado con los incidentes y procesos de impresión, de forma más cómoda que redactado en este Anexo, solo hace falta acceder a la siguiente carpeta:

- [Carpeta con el vídeo explicativo de la impresión 3D.](#)

---

## Anexo 4: Estudio dinámico de una extremidad

A continuación, se encuentran los enlaces a los resultados obtenidos tras el estudio dinámico de una extremidad del robot. De nuevo recordar, que la visualización del movimiento se encuentra disponible en dos vídeos en la nube y los datos disponibles en sus ficheros “xlsx” correspondientes. Estos no son incluidos en esta memoria debido a su extensión individual de 150 páginas por articulación:

- [Enlace al estudio dinámico sin carga.](#)
- [Enlace al estudio dinámico con carga de 30 N.](#)
- [Enlace a los movimientos impuestos y movimientos.](#)
- [Enlace al vídeo de la captura de datos en tiempo real.](#)
- [Enlace al vídeo acelerado x50.](#)

---

## Anexo 5: Estudio de tensiones de una extremidad

A continuación, se encuentran los enlaces a los dos estudios de tensiones sobre el sistema articulado. El primero presenta una carga directa sobre el eje de soporte del servomotor y el segundo sobre el centro de masas de la estructura de soporte del hombro. De nuevo, debido a una extensión individual por fichero de 50 páginas, no se incluyen directamente en la memoria y son accesibles mediante los siguientes enlaces:

- [Enlace al estudio de tensiones con la carga directa sobre el eje de soporte del servomotor.](#)
- [Enlace al estudio de tensiones con la carga sobre el centro de masas de la estructura de soporte del hombro.](#)
- [Enlace al vídeo de generación del estudio de tensiones.](#)